House Price IASD4

BRUNET Olivier, GUEUKAM Raphaël, KACHKACHI Slim, LEFEVRE Benjamin,

TABLE DES MATIERES

#1. INTRODUCTION

L'objectif de cette étude est de proposer un modèle de prédiction de prix de vente de maisons (la 'target') basé sur quelques 67 critères d'évaluation (cf. détails en annexe 3).

Les prédictions utiliseront des modèles de régression linéaire multiple dont la performance sera évaluée sur la base de la plus petite "RMSE" ('Root Mean Square Error'). Le jeu de données comprend une partie 'train' pour l'élaboration des modèles prédictifs et une partie "test" pour l'évaluation de leurs performances.

Concernant les modèles de prédiction, les approches suivantes sont présentées :

1) Utilisation de toutes les variables ;
2) Sélection des variables issue des études de corrélation 'corrplot' ;
3) Sélection des variables issue du 'Random Forest' ;
4) Transformation associée à de la sélection de variables.

NB : deux autres approches testées sont présentées en annexes.
NB : pour assurer la reproductibilité des résultats, le générateur de variables aléatoires est positionné à 2010.

Ce document comprend les parties suivantes :

- Analyse exploratoire des données ;
- Construction et d'évaluation des modèles ;
- Sélection du meilleur modèle ;
- Conclusion et de réflexion sur les axes d'amélioration des prédictions.

Pour des raisons de lisibilité, seuls les 1^{iers} cas des différentes étapes seront détaillés, le reste sera détaillé en annexe.

#2. ANALYSE EXPLORATOIRE/MODELISATION

Dans ce paragraphe après une rapide prise en main des données, nous identifierons :
- La loi de comportement de la 'target' ;
- Les valeurs surprenantes, les pistes de transformation de variables, de regroupement de modalités ;
- Les variables explicatives à forte corrélation avec la loi de comportement de la 'target'.

##2.1 PRISE EN MAIN DES DONNEES

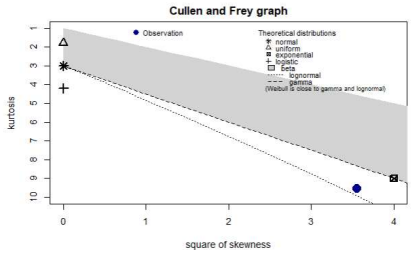
```
#Vérifications préliminaires : absence de doublon ou  de valeurs à "NaN, taux  de zéros etc.
nrow(df_all) - nrow(unique(df_all)) # nbre total de lignes - nbre de lignes apparaissant une fois
any(is.na(df_all))  # présence de na ?
x=df_status(df_all)$p_zeros # variables avec un % de zéros élevés
```

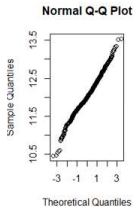
variable <chr>	q_zeros <int>	p_zeros <dbl>	q_na <int>	p_na <dbl>	q_inf <int>	p_inf <dbl>	type <chr>	unique <int>
GarageQual	0	0.00	0	0	0	0	character	5
GarageCond	0	0.00	0	0	0	0	character	5
PavedDrive	0	0.00	0	0	0	0	character	3
WoodDeckSF	761	52.12	0	0	0	0	integer	274
OpenPorchSF	656	44.93	0	0	0	0	integer	202
EnclosedPorch	1252	85.75	0	0	0	0	integer	120

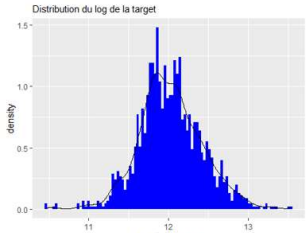
Illustration % de zéros, type des variables et nombre de valeurs

##2.2 DENTIFICATION DE LA LOI DE DISTRIBUTION DE LA TARGET

```
#Le graphe de Cullen-Frey indique une distribution 'lognormale' de la 'target'
options(repr.plot.width = 4, repr.plot.height = 4);descdist(df_all$SalePrice)
#Une autre méthode de vérification a été testée confirmant cette analyse (détaillée en annexe 1).
```





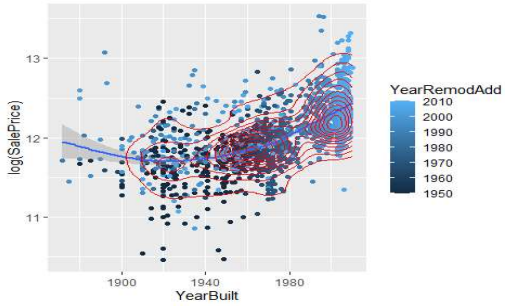


##2.3 ANALYSE DES VARIABLES EXPLICATIVES

Dans cette partie nous mènerons quelques statistiques sur les variables afin d'identifier les pistes de transformation (regroupement de modalités, de variables) voire des points surprenants.

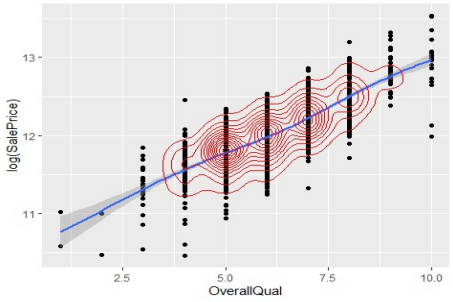
###2.3.1 VARIABLES NUMERIQUES

```
#Commençons par la distribution du 'log(SalePrice)' par rapport à 'YearBuild' (année de construction)
ggplot(df_all, aes(x = YearBuilt, y =log(SalePrice),color=YearRemodAdd)) + geom_point()+geom_smooth()+geom_density2d(color = "red")
```

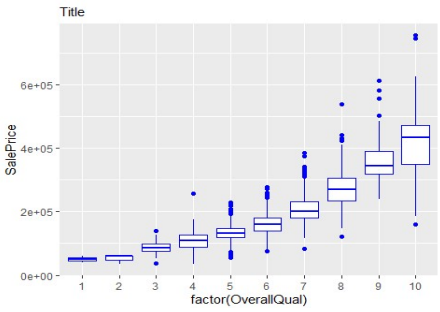


La distribution est linéaire seulement à partir de 1940 ; mais il est difficile de justifier le retrait de points antérieurs à cette date.
A noter que la variable 'GarageYrBlt' se comporte de façon similaire (cf. annexe5).

```
#La variable 'OverallQual' présente une distribution 'linéaire' discontinue avec la 'target'.  A noter un i ntervalle interquartile plus important pour les niveaux de qualité élevée.
```

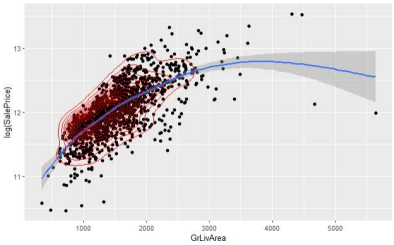


Distribution par rapport au log(SalePrice)

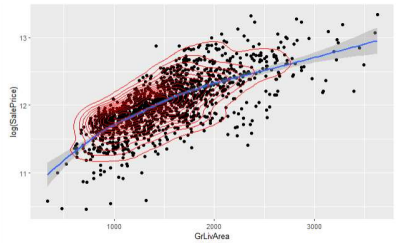


Intervalles interquartiles par modularité

#La distribution de 'GrLivArea' (surface habitable hors sous-sol) présente des points surprenants avec des prix bas pour des grandes surfaces. On tester un modèle sans cette valeur.



Distribution par rapport au log(SalePrice)



Distribution une fois les points extrêmes retirés

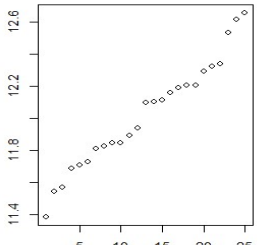
A noter que d'autres variables présentent des points surprenants (liste en annexe 6).

###2.3.2 VARIABLES QUALITATIVES : TRAVAIL SUR LA REDUCTION DU NOMBRE DE MODALITES

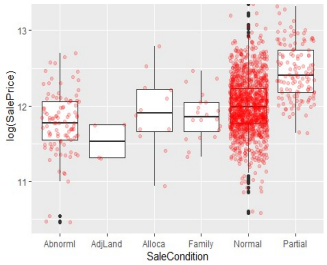
On s'intéressera à certaines variables dans la perspective d'un "feature engineering", car pour certaines d'entre elles, il semble pertinent de réduire leurs modalités voire de les fusionner.

```
#Avec pour la variable 'Neighborhood' ce qui apparait comme des paliers
plot(sort(tapply(log(df_all$SalePrice),df_all$Neighborhood,median)))

#Et pour la variable 'SaleConditions, des valeurs médianes similaires par condition de vente
ggplot(df_all) +geom_boxplot(aes(x = SaleCondition, y = log(SalePrice))) +  geom_jitter(aes(x =SaleCondition, y =log(SalePrice)),col = "red", alpha = 0.2  )
```

'Neighborhood'



'SaleConditions'



La même approche est envisageable pour 'BsmtFinType1' et 'GarageFinish' (cf. annexe 7).

Enfin, certaines variables (cf. annexe 3) comme 'ConditionS' présentent des modalités différentes entre les deux jeux, ce qui s'est avéré bloquant pour les prédictions. On veillera à les retirer lors de l'élaboration des modèles.

###2.3 TRAVAIL SUR LE REGROUPEMENT DE VARIABLES

#On a pu constater que le regroupement des variables de surface 'GrLivArea', 'TotalBsmtSF' (cf. annexe 4) améliore leur taux de corrélation avec la 'target'. On testera l'impact sur un modèle.

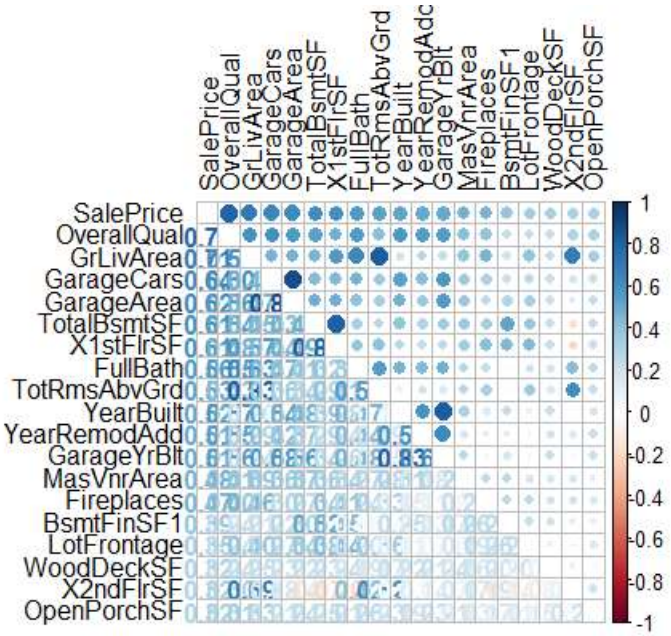
```
cor(df_all$SalePrice, df_all$GrLivArea, use= "pairwise.complete.obs")
## [1] 0.7086245
```

```
cor(df_all$SalePrice, df_all$TotalBsmtSF, use= "pairwise.complete.obs")
## [1] 0.6135806
cor(df_all$SalePrice, df_all$GrLivArea + df_all$TotalBsmtSF, use= "pairwise.complete.obs")
## [1] 0.7789588
```

##2.4 ANALYSE DES CORRELATIONS

#Cherchons les variables quantitatives avec une valeur de corrélation absolue > 0,3.
A noter qu'un essai avec un indice à 0,1 a donné la même liste.

```
numericVars <- which(sapply(df_all, is.numeric));  # index des vecteurs de variables numériques
numericVarNames <- names(numericVars)
all_numVar <- df_all[, numericVars] # nombre length(numericVars)
cor_numVar <- cor(all_numVar, use="pairwise.complete.obs")  # correlations de toutes les variables num.
options(repr.plot.width = 22, repr.plot.height = 11)
cor_sorted <- as.matrix(sort(cor_numVar[,'SalePrice'], decreasing = TRUE)) # classement par ordre décroissant des corrélations
CorHigh <- names(which(apply(cor_sorted, 1, function(x) (x > 0.3 | x < -0.3)))) # filtre pour ne garder que les corrélations  > | 0.3|
cor_numVar <- cor_numVar[CorHigh, CorHigh]
corrplot.mixed(cor_numVar, tl.col="black", tl.pos = "lt") # affichage du graphe
```



Remarque sur les variables fortement corrélées entre elles

De telles variables peuvent dégrader les modèles, car l'hypothèse centrale de la régression linéaire est l'indépendance entre elles. On testera des modèles sans celles-ci :

- 'GarageYrBlt' (année construction garage) avec 'YearBuilt' (AnnéeConstruction) (taux de.83)
- 'TotRmsAbvGrd' (NbrePieces) avec 'GrLivArea' (Surface totale) (taux de corrélation 0.83)
- 'X1stFlrSF' (SurfaceRdC) avec 'TotalBsmtSF' (Surface SSol) (taux de corrélation 0.83)
- 'GarageCars' (NbrePlacesParking) avec 'GarageArea' (Surfacegarage) (taux de corrélation 0.88)
- 'X2ndFlrSF' (Surface 2ieme étage) avec 'GrLivArea' (SurfaceTot) (taux de corrélation 0.69.

#3. MODELISATION

Dans cette partie nous testerons plusieurs modèles. La RMSE sera utilisée pour évaluer leur performance alors que leur validité sera evaluée en vérifiant les hypothèses suivantes :

- [P1] : Centrage des erreurs ;
- [P2] : Variance homoscédastique ;
- [P3] : Non corrélation des erreurs ;
- [P4] : Distribution gaussiennes des erreurs.

##3.1. MODELE DE BASE : REGRESSION LINEAIRE A PARTIR DE TOUTES LES VARIABLES

```
#Etape 1 : dans la prise en main des données, nous avions identifié quelques variables à retirer
train = subset(df_train, select=-c(Condition2, RoofMatl, Exterior1st, Exterior2nd, Heating, Electrical))
testt = subset(df_testt, select=-c(Condition2, RoofMatl, Exterior1st, Exterior2nd, Heating, Electrical))

#Etape 2 : définition du 1ier modèle réduit à l'intercept
model_base = lm(log(train$SalePrice)~., data=train)
summary(model_base)
```

Notons un 'Adjustet R_squared' proche de1, donc un modèle qui colle aux données.

```
SaleConditionNormal   5.519e-02  1.716e-02   3.217 0.001341 **
SaleConditionPartial -9.379e-02  9.933e-02  -0.944 0.345294
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.1225 on 917 degrees of freedom
Multiple R-squared:  0.9187,    Adjusted R-squared:  0.903
F-statistic: 58.56 on 177 and 917 DF,  p-value: < 2.2e-16
```

```
#Etape 3 :  identification puis suppression des 'outliers' (et points surprenants identifiés au départ)
outlierTest(model_base)
train_b = train[-c(720,819,743,661,814,1079),]
train_b = subset(train_b, GrLivArea < 4000) ;train_b = subset(train_b, GarageArea < 1250) ;train_b = subset(train_b, TotalBsmtSF < 4000),train_b = subset(train_b, X1stFlrSF < 4500) ;train = subset(train_b, MasVnrArea < 1000)

#Etape 4 : création d'un nouveau modèle sans les 'outliers' et points extrêmes
model_baseb = lm(log(SalePrice)~., data=train_b)
outlierTest(model_baseb)

#Etape 5 : retrait d'un 2ieme série des 'outliers'
train_c = train_b[-c(359,1002,794,792,749,3,994),]
model_basec = lm(log(SalePrice)~., data=train_c)

#Etape 6 : Comparaison des AIC
AIC=c(extractAIC(model_base)[2],extractAIC(model_baseb)[2],extractAIC(model_basec)[2])
names(AIC)=c('modèle_base','model_base_sans_outliers1','model_base_sans_outliers2')
```

Modèles	Référence	Retrait 1iere série d'outliers	Retrait 2ieme série d'outliers
AIC	-4437.287	-4987.321	-4972.251

Analyse de validité du modèle retenu (modèle sans la 1iere série d'outliers)

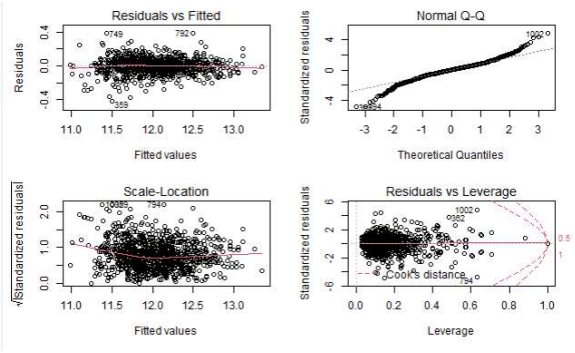
[P1] (plot 'Residuals vs Fitted') est validée car :
- Les résidus restent globalement uniformément répartis des deux côtés de 0
- La ligne rouge est approximativement horizontale à zéro

[P2] (plot 'Scale-Location') semble validée avec une courbe rouge quasi horizontale et des points uniformément répartis ; pour autant le test de Breush-Pagan donne un p-value <0,05 (10^{-11}) ;

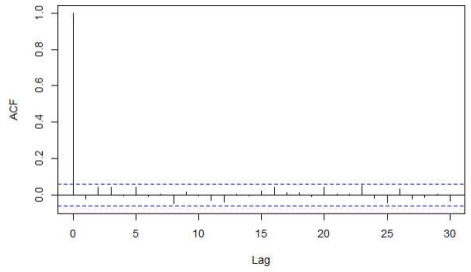
[P3] (plot 'Auto-corrélation') est validée car tous les traits verticaux ne dépassent pas les seuils en pointillé (à l'exception du 1ler) ;

[P4] (plot 'Normal Q-Q') est difficile à valider, les points semblant presque tous alignés autour de la 1iere bissectrice. Le test de Shapiro invalide finalement cette hypothèse (p-value <0,05 /10^{-14}).

```
plot(model_baseb)
```




```
acf(residuals(model_baseb), main="Plot Auto-corrélation")
```




```
ncvTest(model_baseb)
```

```
Non-constant Variance Score Test
Variance formula: ~ fitted.values
Chisquare = 42.60578, Df = 1, p = 6.6961e-11
```

```
shapiro.test(residuals(model_baseline)
```

```
        Shapiro-Wilk normality test

data:  residuals(model_baseline)
W = 0.90252, p-value < 2.2e-16
```

Calcul du RMSE

```
# calcul des prédictions pour le train & le test
y_train_pred = (predict(model_baseb, newdata=train_b)) ; y_testt_pred = (predict(model_baseb, newdata=testt))

# calcul des RMSE: l'exponentielle est appliqué pour annuler le log
RMSE_train = c(sqrt(mean((exp(y_train_pred)-train_b$SalePrice)^2))) ;
RMSE_testt = c(sqrt(mean((exp(y_testt_pred)-testt$SalePrice)^2)))
# affichage des valeurs
print("RMSE sur le dataset de train:"); print(RMSE_train, digits=5)
## [1] "RMSE sur le dataset de train:"
## [1] 15 897
print("RMSE sur le dataset de test:"); print(RMSE_testt, digits=5)
## [1] "RMSE sur le dataset de test:"
## [1] 21 579
```

3.2 Modèle construit à partir des variables sélectionnées par le 'corrplot'

Seuls les résultats sont présentés ; les formules sont présentées en annexe 8.

Etapes	
Retrait des variables dont les modalités diffèrent entre train et test	Condition2, RoofMatl, Exterior1st, Exterior2nd, Heating, Electrical
Sélection des variables (taux de corrélation >0,3) sans les variables fortement corrélées à d'autres	OverallQual, GrLivArea, GarageCars, TotalBsmtSF, FullBath, YearBuilt, YearRemodAdd, MasVnrArea, BsmtFinSF1, WoodDeckSF, OpenPorchSF, SalePrice
Retrait des outliers	*720,819,743,585*

Comparaison des AIC afin de choisir le modèle qui sera étudié plus en avant

Modèles	Référence	Retrait des outliers
AIC	-3912.957	-4342.930

Analyse de validité (sélection des variables issues du 'corrplot')

	Référence	Sans la 1iere série d'outliers	Rappel cas avec toutes les variables sans 'outliers'
P[1]	Validée	Validée	Validée
P[2]	Non validée p value 10^{-16}	Non validée p value 10^{-6}	Non validée p value 10^{-11}
P[3]	Validée	Validée	Validée
P[4]	Non validée p value 10^{-16}	Non validée p value 10^{-15}	Non validée p value 10^{-11}

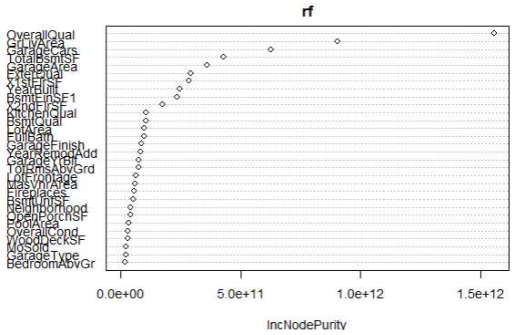
Calcul du RMSE du modèle issu des variables sélectionnées par le corrplot

RMSE sur le dataset de train : 24 560
RMSE sur le dataset de test : 27 370

3.3 Modèle construit à partir des variables sélectionnées par le 'Random Forest'

Etapes	
Retrait des variables dont les modalités diffèrent entre train et test	Condition2, RoofMatl, Exterior1st, Exterior2nd, Heating, Electrical
Identification des variables de plus haute importance	rf = randomForest(SalePrice~ .,data=train) varImpPlot(rf)
Sélection des variables et construction du modèle de référence (sans les variables corrélées à d'autres)	OverallQual ,GrLivArea, Neighborhood, GarageCars, ExterQual, TotalBsmtSF, GarageArea, BsmtFinSF1, KitchenQual, YearBuilt,, BsmtQual, LotArea, FullBath , YearRemodAdd, LotFrontage, BsmtUnfSF, MasVnrArea, Fireplaces, OpenPorchSF, BsmtFinType1, WoodDeckSF, OverallCond, PoolArea, SaleCondition, BsmtExposure ,MoSold, SalePrice
1iere série de retrait des 'outliers' et nouveau modèle	720,819,743,661,1079,814,404
2ieme série de retrait des outliers et nouveau modèle	794, 749

Illustration des variables sélectionnées par le 'Random Forest' par niv. d'importance



Comparaison des AIC afin de choisir le modèle qui sera étudié plus en avant

Modèle	Référence	Retrait 1^{iere} série d' outliers	Retrait 2^{ieme} série d' outliers
AIC	-4290.297	-4846.310	-4837.826

####Analyse de validité du modèle retenu (celui avec la 1iere série des 'outliers' retirés)

	Référence	Sans la 1iere série d'outliers	Sans la 2ieme série d'outliers
P[1]	Validée	Validée	Validée
P[2]	Non validée p value 10^{-16}	Non validée p value 10^{-7}	Non validée p value 10^{-7}
P[3]	Validée	Validée	Validée
P[4]	Non validée p value 10^{-16}	Non validée p value 10^{-13}	Non validée p value 10^{-13}

Calcul du RMSE modèle issu du Random Forest

RMSE sur le dataset de train : 18 175
RMSE sur le dataset de test : 21 692

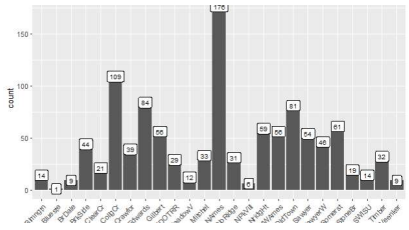
3.4. Recherche d'une approche "Feature engineering" avec transformation de variables

3.4.1 Travail sur la variable 'Neighboorhood', la plus importante selon le 'random forest'

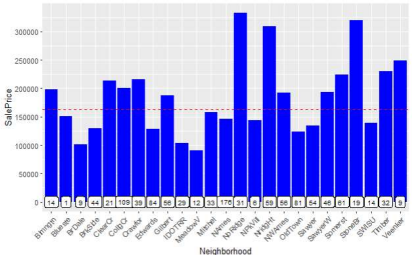
Regardons la fréquence (nombre de maisons) pour chaque mode de 'Neighorhood', puis comparons les valeurs médianes du « SalePrice » pour chacune des modalités.

```
par(mfrow=c(2,2))
options(repr.plot.width = 22, repr.plot.height = 6)
ggplot(data=df_train, aes(x=Neighborhood)) + geom_histogram(stat='count')+ geom_label(stat = "count", aes(label = ..count.., y = ..count..), size=3)+ theme(axis.text.x = element_text(angle = 45, hjust = 1))

ggplot(df_train[!is.na(df_train$SalePrice),], aes(x=Neighborhood, y=SalePrice)) + geom_bar(stat='summary', fun.y = "median", fill='blue') + theme(axis.text.x = element_text(angle = 45, hjust = 1)) + scale_y_continuous(breaks= seq(0, 800000, by=50000)) + geom_label(stat = "count", aes(label = ..count.., y = ..count..), size=3) + geom_hline(yintercept=163000, linetype="dashed", color = "red")
```

Fréquence par modalité



Et valeurs médianes SalePrice



```
# Rassemblons les modes de cette variable‘en 3 valeurs ordinales :

train_var = subset(df_train, select=-c(Condition2, RoofMatl, Exterior1st, Exterior2nd, Heating, Electrical))
test_var = subset(df_testt, select=-c(Condition2, RoofMatl, Exterior1st, Exterior2nd, Heating, Electrical))
train_var$NeighRich[train_var$Neighborhood %in% c('StoneBr', 'NridgHt', 'NoRidge')] <- 2
test_var$NeighRich[test_var$Neighborhood %in% c('StoneBr', 'NridgHt', 'NoRidge')] <- 2
train_var$NeighRich[!train_var$Neighborhood %in% c('MeadowV', 'IDOTRR', 'BrDale', 'StoneBr', 'NridgHt', 'NoRidge')] <- 1
test_var$NeighRich[!test_var$Neighborhood %in% c('MeadowV', 'IDOTRR', 'BrDale', 'StoneBr', 'NridgHt', 'NoRidge')] <- 1
train_var$NeighRich[train_var$Neighborhood %in% c('MeadowV', 'IDOTRR', 'BrDale')] <- 0
test_var$NeighRich[test_var$Neighborhood %in% c('MeadowV', 'IDOTRR', 'BrDale')] <- 0
train_var = (subset(train_var, select=-c(Neighborhood)))
test_var = (subset(test_var, select=-c(Neighborhood)))
```

###3.4.2 Travail sur les variables de surfaces

```
# Regroupons certaines variables de surfaces (corrélation supérieure une fois rassemblées)
train_var$Area <- train_var$GrLivArea + train_var$TotalBsmtSF
test_var$Area <- test_var$GrLivArea + test_var$TotalBsmtSF
# suppression des anciennes
train_var = (subset(train_var, select=-c(GrLivArea, TotalBsmtSF)))
test_var = (subset(test_var, select=-c(GrLivArea, TotalBsmtSF)))
```

###3.4.3 Construction du modèle et analyse de validité

Etapes	
Sélection de toutes les variables et construction d’un nouveau modèle	*#Etape 1 : modèle avec les deux variables modifiées* model_var = lm(log(SalePrice)~., data=train_var)
1iere série de retrait des ‘outliers’	720,819,743,814,1079
2ieme série de retrait des ‘outliers’	359,749,1002,794,792,661

Comparaison des AIC afin de choisir le modèle le plus intéressant

Modèle	Référence	Retrait 1^{iere} série d’outliers	Retrait 2^{ieme} série d’outliers
AIC	4329.517	-4866.607	-4872.241

Analyse de validité des modèles

	Référence	Sans la 1iere série d’outliers	Sans la 2ieme série d’outliers
P[1]	Validée	Validée	Validée
P[2]	Non validée p value 10^{-12}	Non validée p value 10^{-13}	Non validée p value 10^{-11}
P[3]	Validée	Validée	Validée
P[4]	Non validée p value 10^{-16}	Non validée p value 10^{-13}	Non validée p value 10^{-13}

Calcul du RMSE du modèle issu des variables modifiées

RMSE sur le dataset de train : 16 828
RMSE sur le dataset de test : 21 574

#4. CHOIX DU MODELE FINAL / COMPARAISON DU RMSE

Rappel des différents modèles (avec pour chacun des essais avec ou sans 'outliers') :
1) Pas de sélection des variables ;
2) Sélection sur la base du 'corrplot' (mais sans les variables fortement corrélées à d'autres) ;
3) Sélection issue du 'Random Forest' (mais sans les variables fortement corrélées à d'autres) ;
4) Adaptation du 1^{ier} cas en modifiant les variables 'Neighborhood', 'GrLivArea', 'TotalBsmtSF'.

De façon générale nous constatons pour tous nos modèles que :
- Les hypothèses P[1] et P[3] sont validées contrairement aux hypothèses P[2] et P[4] ;
- les 'R-squared' sont proches de 1 (~0.9) (modèles relativement bien ajustés aux valeurs).

Concernant l'hypothèse P[2], le 2^{ier} et 3^{ieme} modèles donnent la p-value la moins dégradée.
Concernant l'hypothèse P4] aucun modèle n'est satisfaisant. Dans le TP Séoul, il est mentionné que ce test ne fonctionne pas pour les tailles de données importantes.

	Base	'Corrplot'	Random Forest	Transfo Variables
P[2]	10^{-11}	10^{-6}	10^{-7}	10^{-13}
P[4]	10^{-16}	10^{-15}	10^{-13}	10^{-13}

Néanmoins ces données ne sont les seules à considérer seul. Visualisons les RMSE.

##4.1 COMPARAISON DES RMSE

Les RMSE obtenus sur le jeu de TEST sont similaires à l'exception du modèle 'Corrplot'

	Base	'Corrplot'	Random Forest	Transfo Variables
Train	15 897	24 560	18 175	16 828
Test	21 579	27 370	21 692	21 574

##4.2 CHOIX DU MODELE FINAL

Les modèles avec 'outliers' n'ont pas été présentés car ne présentant pas d'intérêt. Pour les autres, les meilleures valeurs de RMSE sur le jeu de test tournent autour de 22 000 (USD) à comparer aux 180 k$ en moyenne (soit ~12% du prix moyen soit une marge de +/- 6%).

D'autres méthodes (données normalisées) ont été évaluées (présentées en annexe 11 et 12,) mais n'ont pas apporté d'améliorations.

De façon générale, nous avons constaté que :

-La réduction du nombre de variables dégradait de façon significative la qualité de la prédiction (c'est particulièrement notable avec la sélection issue du 'corrplot') ;
-La RMSE de test est toujours légèrement supérieure à celle de train (on en déduit que les modèles "n'overfit pas", qu'ils peuvent être utilisés sur d'autres jeux) ;
- Le travail sur la transformation des variables 'Neighborhood', 'AGrLivArea', 'TotalBsmtSF' n'a pas apporté de gain escompté.

En définitive, en tenant des hypothèses de validité des modèles, notre meilleure prédiction est issue de la sélection du 'Random Forest' (y compris pour le test de variance homoscédastique) ; avec une RSME de 18 175 $ sur le train et de 21 672 $ sur le test.

#5 DISCUSSION

Le premier constat est que le 'feature engineering' sur quelques variables n'a pas été déterminant pour la qualité de la prédiction, contrairement à la suppression des 'outliers'.

La validation de l'hypothèse P[2] reste la difficulté sur laquelle nous avons butée.

Pour améliorer nos modèles prédictifs, les pistes par ordre de priorité nous semblent être :

- Poursuivre la création de 'features' (regroupement de variables, réduction de modalités, application de fonctions aux valeurs numériques (valeurs au carré...) ;

- Essayer d'autres modèles notamment ceux de type "ensemblistes" sur la base d'arbre comme le 'Random Forest', moins sensibles aux valeurs aberrantes.

#ANNEXES

1. Décompte des variables numériques avec des taux élevés de zéros
2. Autre méthode pour déterminer la loi de comportement de la 'target'
3. Variables qualitatives dont les modalités sont différentes entre les jeux train et test
4. Illustration de l'amélioration du taux de corrélation en regroupant les variables quantitatives de surfaces
5. Distribution des variables "GarageYrBlt" (année de construction du garage) et "GarageCars" nombre de places de parking) rapport à la Target
6. Distribution des variables "GarageArea" (surface du garage), "1st Floor" (surface du RdC) et "TotalBsmtSF" (surface totale du sous-sol) par rapport à la Target
7. Distribution de la variable "BsmtFinType1" (qualité du sous-sol pour l'aménagement en pièce à vivre) et "GarageFinish" ("état de finition intérieur du garage).
8. Etude de validé du modèle issu de selection des variables par le Corrplot
9. Etude de validé du modèle issu de selection des variables par le Random Forest
10. Etude de validé du modèle issu des transformation des variables "Neighborhood", "AGrLivArea" et "TotalBsmtSF"
11. Modéle sans sélection, sans 'outliers' mais normalisé
12. Modèle mixant modification de variables et sélection en utilisant les p-values de la synthèse du modèle de régression linéairessu d'une
13. Description des différentes variables prédictives

###Annexe 1 décompte des variables numériques avec des zéros

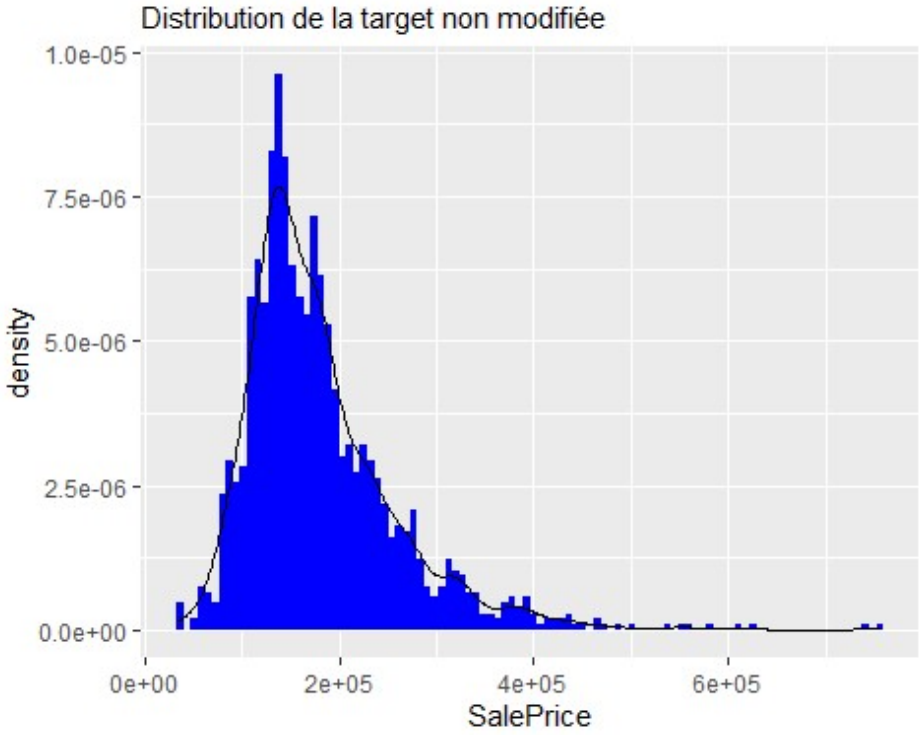
```
x=df_status(df_all)$p_zeros
```

```
##         variable q_zeros p_zeros q_na p_na q_inf p_inf      type unique
## 1     MSSubClass       0    0.00    0    0     0     0   integer     15
## 2       MSZoning       0    0.00    0    0     0     0 character      5
## 3    LotFrontage       0    0.00    0    0     0     0   integer    110
## 4        LotArea       0    0.00    0    0     0     0   integer   1073
## 5         Street       0    0.00    0    0     0     0 character      2
## 6       LotShape       0    0.00    0    0     0     0 character      4
## 7    LandContour       0    0.00    0    0     0     0 character      4
## 8      Utilities       0    0.00    0    0     0     0 character      2
## 9      LotConfig       0    0.00    0    0     0     0 character      5
## 10     LandSlope       0    0.00    0    0     0     0 character      3
## 11  Neighborhood       0    0.00    0    0     0     0 character     25
## 12    Condition1       0    0.00    0    0     0     0 character      9
## 13    Condition2       0    0.00    0    0     0     0 character      8
## 14      BldgType       0    0.00    0    0     0     0 character      5
## 15    HouseStyle       0    0.00    0    0     0     0 character      8
## 16   OverallQual       0    0.00    0    0     0     0   integer     10
## 17   OverallCond       0    0.00    0    0     0     0   integer      9
## 18     YearBuilt       0    0.00    0    0     0     0   integer    112
## 19  YearRemodAdd       0    0.00    0    0     0     0   integer     61
## 20     RoofStyle       0    0.00    0    0     0     0 character      6
## 21      RoofMatl       0    0.00    0    0     0     0 character      8
## 22   Exterior1st       0    0.00    0    0     0     0 character     15
## 23   Exterior2nd       0    0.00    0    0     0     0 character     16
## 24    MasVnrType       0    0.00    0    0     0     0 character      4
## 25    MasVnrArea     868   59.45    0    0     0     0   integer    327
## 26     ExterQual       0    0.00    0    0     0     0 character      4
## 27     ExterCond       0    0.00    0    0     0     0 character      5
## 28    Foundation       0    0.00    0    0     0     0 character      6
## 29      BsmtQual       0    0.00    0    0     0     0 character      4
## 30      BsmtCond       0    0.00    0    0     0     0 character      4
## 31  BsmtExposure       0    0.00    0    0     0     0 character      4
## 32  BsmtFinType1       0    0.00    0    0     0     0 character      6
## 33    BsmtFinSF1     467   31.99    0    0     0     0   integer    637
## 34  BsmtFinType2       0    0.00    0    0     0     0 character      6
## 35    BsmtFinSF2    1293   88.56    0    0     0     0   integer    144
## 36     BsmtUnfSF     118    8.08    0    0     0     0   integer    780
## 37   TotalBsmtSF      37    2.53    0    0     0     0   integer    721
## 38       Heating       0    0.00    0    0     0     0 character      6
## 39     HeatingQC       0    0.00    0    0     0     0 character      5
## 40    CentralAir       0    0.00    0    0     0     0 character      2
## 41    Electrical       0    0.00    0    0     0     0 character      5
## 42     X1stFlrSF       0    0.00    0    0     0     0   integer    753
## 43     X2ndFlrSF     829   56.78    0    0     0     0   integer    417
## 44  LowQualFinSF    1434   98.22    0    0     0     0   integer     24
## 45     GrLivArea       0    0.00    0    0     0     0   integer    861
## 46  BsmtFullBath     856   58.63    0    0     0     0   integer      4
## 47  BsmtHalfBath    1378   94.38    0    0     0     0   integer      3
## 48      FullBath       9    0.62    0    0     0     0   integer      4
```

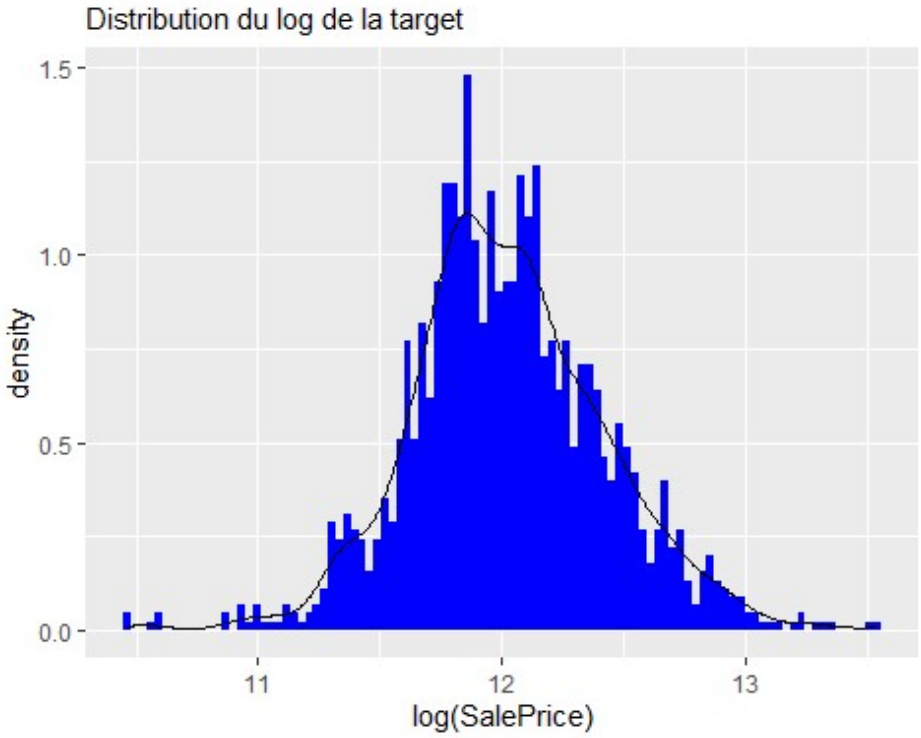
```
## 49      HalfBath     913   62.53    0    0     0     0   integer      3
## 50  BedroomAbvGr       6    0.41    0    0     0     0   integer      8
## 51  KitchenAbvGr       1    0.07    0    0     0     0   integer      4
## 52   KitchenQual       0    0.00    0    0     0     0 character      4
## 53  TotRmsAbvGrd       0    0.00    0    0     0     0   integer     12
## 54    Functional       0    0.00    0    0     0     0 character      7
## 55    Fireplaces     690   47.26    0    0     0     0   integer      4
## 56    GarageType       0    0.00    0    0     0     0 character      6
## 57   GarageYrBlt       0    0.00    0    0     0     0   integer     97
## 58  GarageFinish       0    0.00    0    0     0     0 character      3
## 59    GarageCars      81    5.55    0    0     0     0   integer      5
## 60    GarageArea      81    5.55    0    0     0     0   integer    441
## 61    GarageQual       0    0.00    0    0     0     0 character      5
## 62    GarageCond       0    0.00    0    0     0     0 character      5
## 63    PavedDrive       0    0.00    0    0     0     0 character      3
## 64    WoodDeckSF     761   52.12    0    0     0     0   integer    274
## 65   OpenPorchSF     656   44.93    0    0     0     0   integer    202
## 66 EnclosedPorch    1252   85.75    0    0     0     0   integer    120
## 67    X3SsnPorch    1436   98.36    0    0     0     0   integer     20
## 68   ScreenPorch    1344   92.05    0    0     0     0   integer     76
## 69      PoolArea    1453   99.52    0    0     0     0   integer      8
## 70       MiscVal    1408   96.44    0    0     0     0   integer     21
## 71        MoSold       0    0.00    0    0     0     0   integer     12
## 72        YrSold       0    0.00    0    0     0     0   integer      5
## 73      SaleType       0    0.00    0    0     0     0 character      9
## 74 SaleCondition       0    0.00    0    0     0     0 character      6
## 75     SalePrice       0    0.00    0    0     0     0   integer    663
```

###Annexe 2 Loi de comportement de la Target

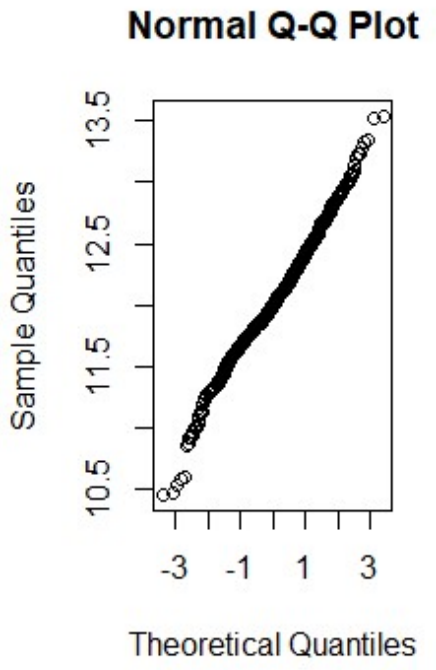
```
#Deuxième méthode pour vérifier la distribution log normale de la target
par(mfrow=c(1,2))
options(repr.plot.width = 2, repr.plot.height = 2)
ggplot(data = df_all[!is.na(df_all$SalePrice),], aes(x=SalePrice)) +
        geom_histogram(bins=100, fill="blue", aes(y = ..density..)) +
        geom_density() + labs(subtitle="Distribution de la target non modifiée
")
```




```
ggplot(data=df_all[!is.na(df_all$SalePrice),], aes(x=log(SalePrice))) +
        geom_histogram(bins=100, fill="blue", aes(y = ..density..)) +
        geom_density() + labs(subtitle="Distribution du log de la target")
```




```
qqnorm(log(df_all$SalePrice))
```



###Annexe 3 Variables qualitatives dont les modalités sont différentes entre les deux jeux

Exterior1st" et "Exterior2nd" (type de matériaux sur les murs, 1ier et 2ieme matériaux),"RoofMatl" (matériaux de la toiture), "Heating" (type de chauffage), "Electrical" (information sur le système électrique), "ConditionS" (proximité aux facilités dans le cas de plus d'une facilité)

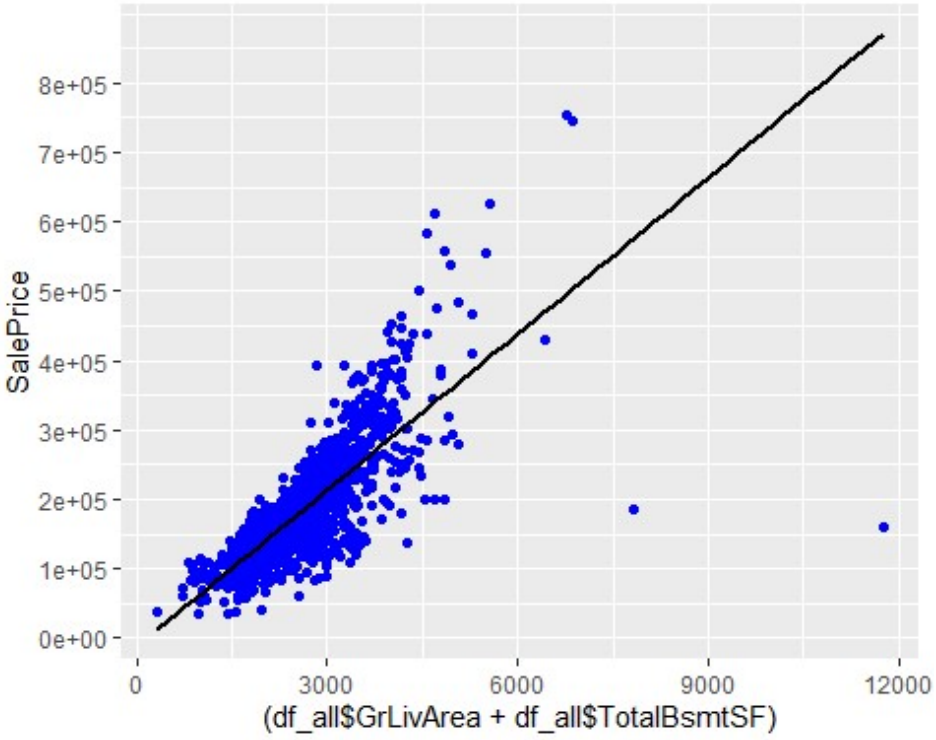
Illustration sur la variable "RoofMatl"

```
table(df_train$RoofMatl)

##
## ClyTile CompShg   Metal    Roll Tar&Grv WdShake WdShngl
##       1    1078       1       1       6       5       3

table(df_testt$RoofMatl)

##
## CompShg Membran Tar&Grv WdShngl
##     356       1       5       3
```

###Annexe 4 Illustration de l'amélioration du taux de corrélation en regroupant les variables quantitatives de surfaces Cette partie démontre que la totalité des surfaces ont une plus grande influence sur le prix. Néanmoins, ici nous n'avons pas à créer une nouvelle feature avec la somme des différentes surfaces car c'est implicitement fait par la régression linéaire (qui fait toute sorte de combinaison linéaire)

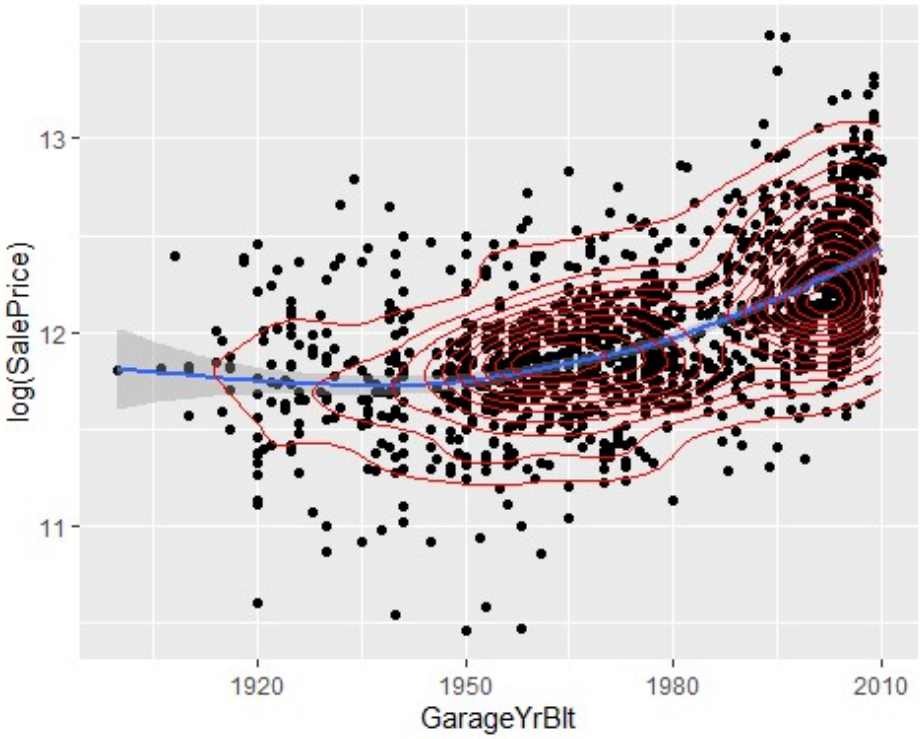
```
# taille des figures
options(repr.plot.width = 5, repr.plot.height = 3)

ggplot(data=df_all[!is.na(df_all$SalePrice),], aes(x=(df_all$GrLivArea + df_al
```

```
l$TotalBsmtSF), y=SalePrice))+
        geom_point(col='blue') + geom_smooth(method = "lm", se=FALSE, color="b
lack", aes(group=1)) +
        scale_y_continuous(breaks= seq(0, 800000, by=100000))

## `geom_smooth()` using formula 'y ~ x'
```



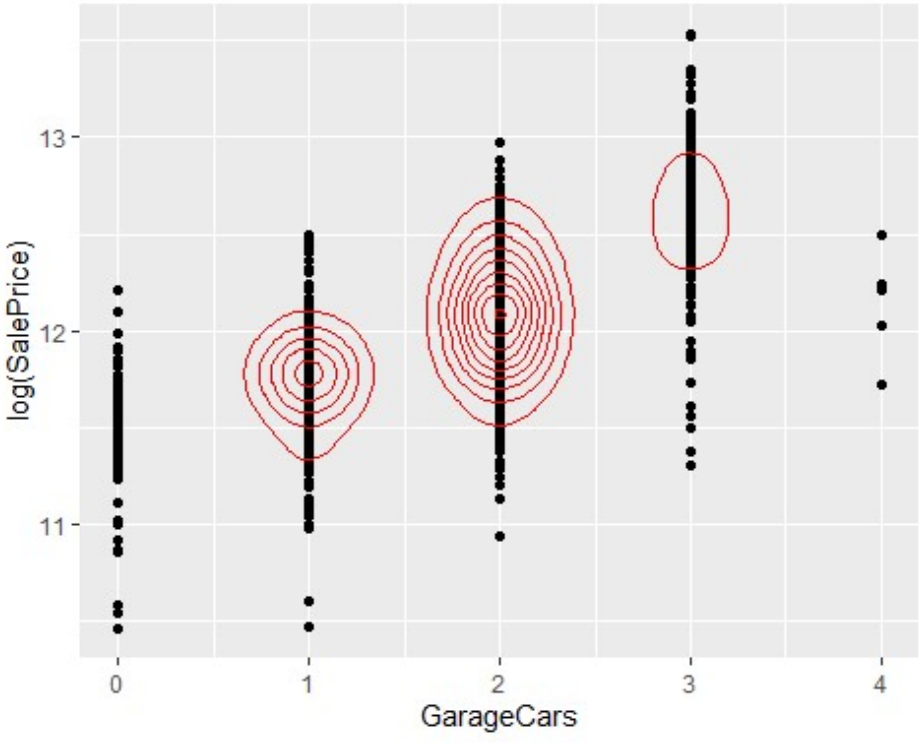
###Annexe 5 Distribution des variables "GarageYrBlt" (année de construction du garage) et "GarageCars" nombre de places de parking) rapport à la Target

```
#Distribution linéaire de l'année de construction du garage par rapport au log
de la 'target' à partir des année 50
ggplot(df_all, aes(x = GarageYrBlt, y =log(SalePrice))) + geom_point()+geom_sm
ooth()+geom_density2d(color = "red")

## `geom_smooth()` using method = 'gam' and formula 'y ~ s(x, bs = "cs")'
```




```
#distribution linéaire du nombre de places de parking par rapport au log de la target même s'il existe quelques points surprenants
ggplot(df_all,aes(x = GarageCars, y =log(SalePrice))) + geom_point()+geom_smooth()+geom_density2d(color = "red")
```

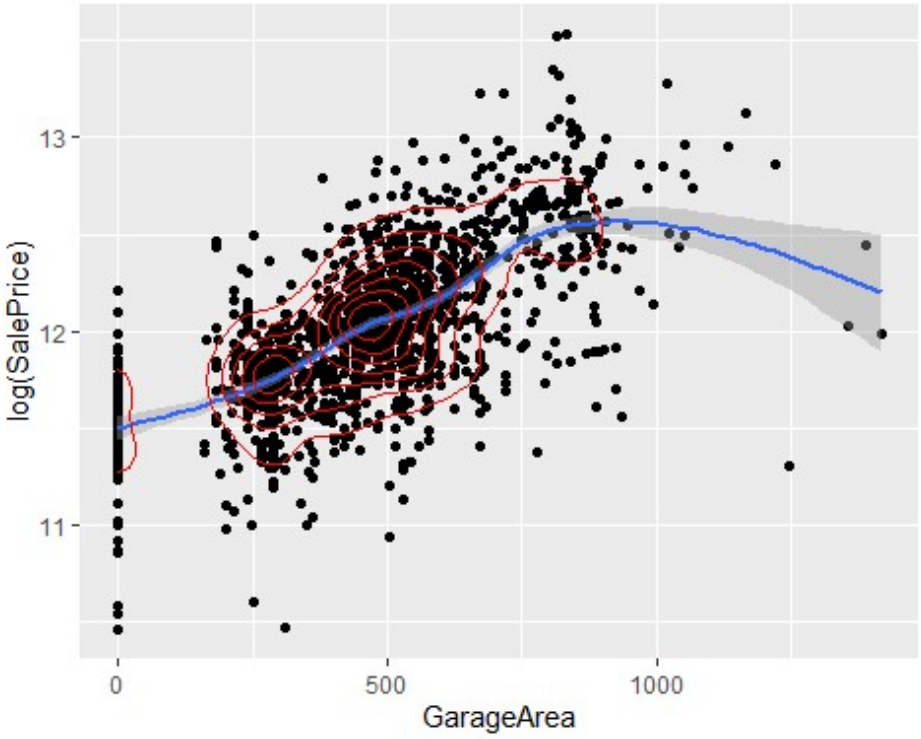
```
## `geom_smooth()` using method = 'gam' and formula 'y ~ s(x, bs = "cs")'
```



###Annexe 6 Distribution des variables “GarageArea” (surface du garage), “1st Floor” (surface du RdC) et “TotalBsmtSF” (surface totale du sous-sol) par rapport à la Target

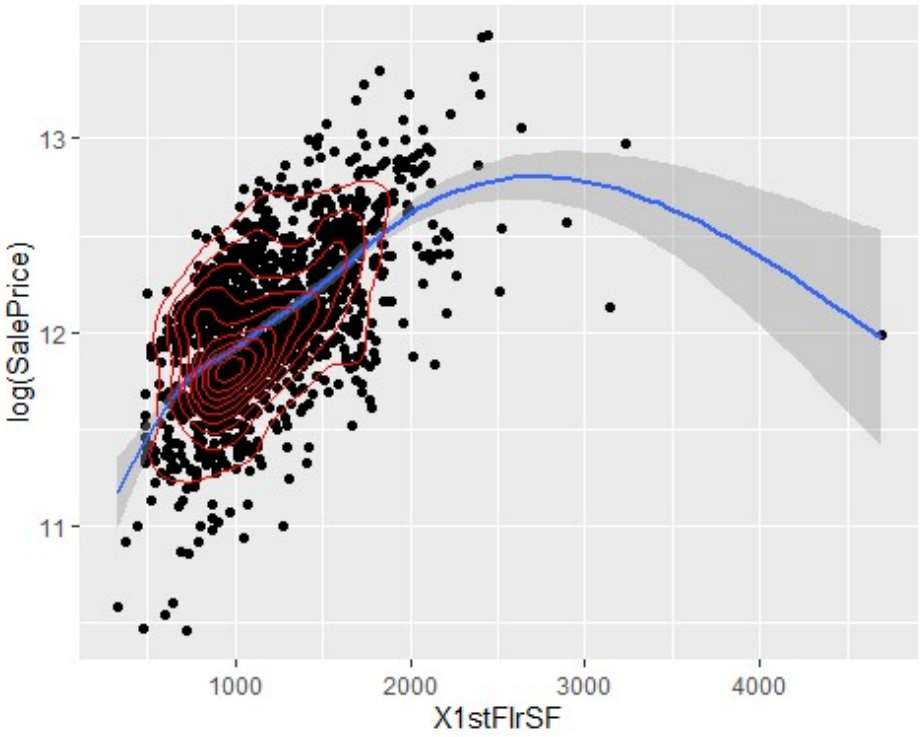
```
#De façon analogue on peut retirer quelques points pour les surface garage au-delà de 1500 sq feet avec des prix anormalement bas.
ggplot(df_all,aes(x = GarageArea, y =log(SalePrice))) + geom_point()+geom_smoo
th()+geom_density2d(color = "red")
```

```
## `geom_smooth()` using method = 'gam' and formula 'y ~ s(x, bs = "cs")'
```

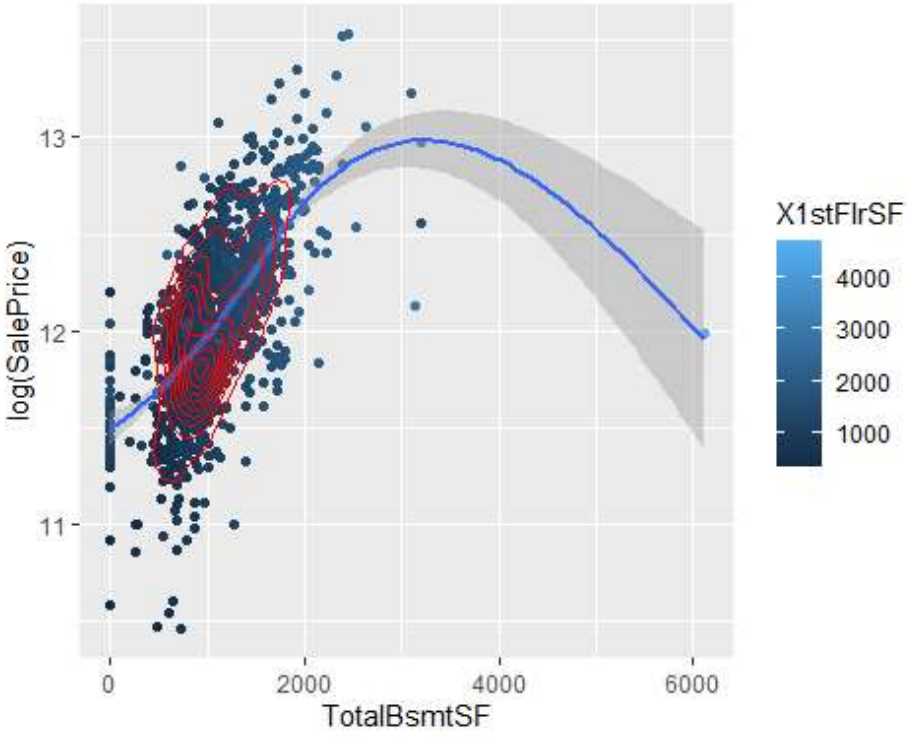



```
#Ici on retrouve clairement un point qui sort du lot : la surface de "1st Floo
r" au dela de 4000 sq. feet.
#Sans ce point la modélisation tend plus vers une droite: il serait intéressan
t de retirer ce  point extreme car le prix est également très bas.
ggplot(df_all,aes(x = X1stFlrSF, y =log(SalePrice))) + geom_point()+geom_smoot
h()+geom_density2d(color = "red")
```

```
## `geom_smooth()` using method = 'gam' and formula 'y ~ s(x, bs = "cs")'
```

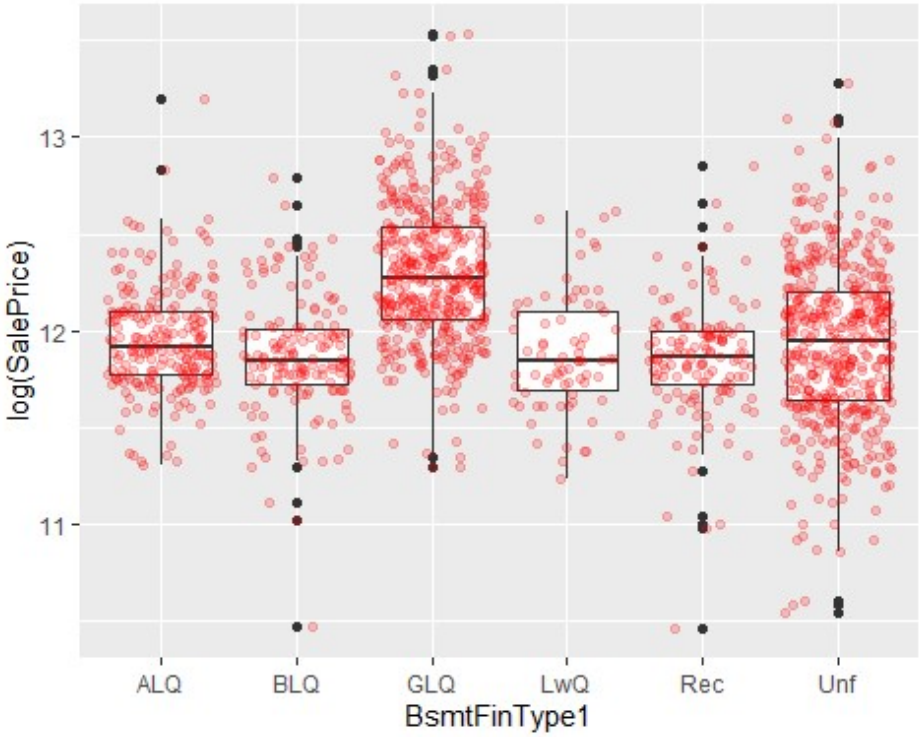



```
#Meme chose pour le basement (sous-sol) : un point extrême se dégage, il s'agi
t de point très excentré par rapport à l'ensemble des valeurs
ggplot(df_all,aes(x = TotalBsmtSF, y =log(SalePrice),color=X1stFlrSF)) + geom_
point()+geom_smooth()+geom_density2d(color = "red")

## `geom_smooth()` using method = 'gam' and formula 'y ~ s(x, bs = "cs")'
```

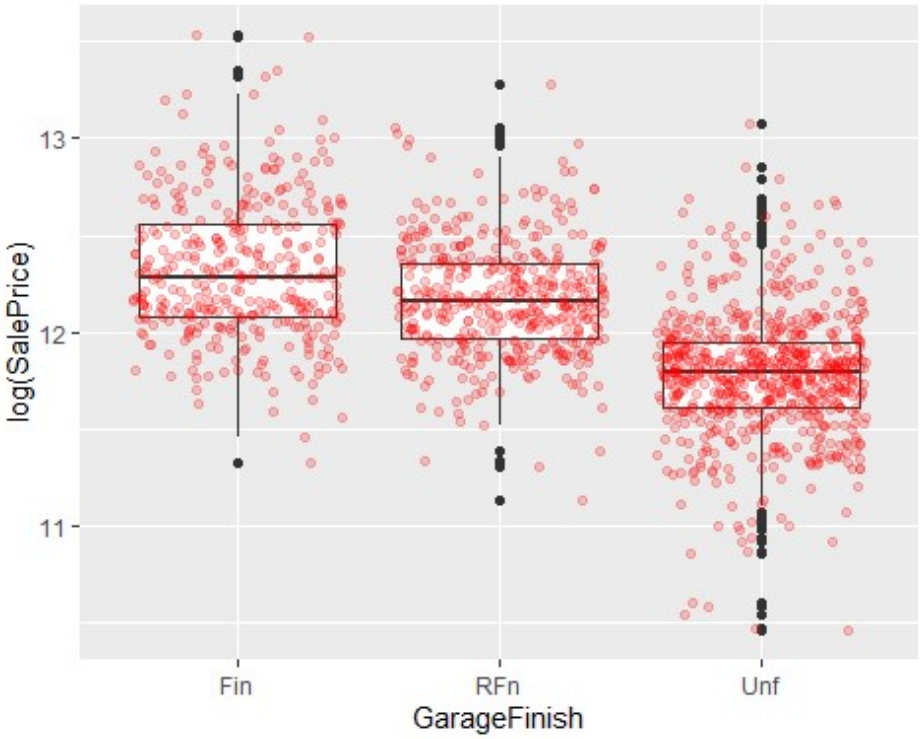


###Annexe 7 Distribution de la variable “BsmtFinType1” (qualité du sous-sol pour l’aménagement en pièce à vivre) et “GarageFinish” (“état de finition intérieur du garage).

```
#des modalités que l'on peut regrouper: toutes sauf GLQ (Good Level Quarter) c
e qui a un sens car ttes les autres correspondent à des niv de qualité inférie
ure
ggplot(df_all) +
  geom_boxplot(aes(x = BsmtFinType1, y = log(SalePrice)))+
  geom_jitter(
    aes(x = BsmtFinType1, y = log(SalePrice)),
    col = "red", alpha = 0.2
  )
```




```
#3 modalités dont deux ont des valeurs médianes très proches/ on doit pouvoir
regrouper Fin et RFn
ggplot(df_all) +
  geom_boxplot(aes(x = GarageFinish, y = log(SalePrice)))+
  geom_jitter(
    aes(x = GarageFinish, y = log(SalePrice)),
    col = "red", alpha = 0.2
  )
```



###Annexe 8 étude de validé du modèle issu de selection des variables par le corrplot

###A8 visualisation des outliers issus du test de Bonferonni

```
outlierTest(model_cor)
```

```
##       rstudent unadjusted p-value Bonferroni p
## 720 -16.533356         6.8155e-55   7.4630e-52
## 819  -9.414525         2.7862e-20   3.0508e-17
## 743  -5.303150         1.3808e-07   1.5119e-04
## 585  -4.471735         8.5809e-06   9.3961e-03
```

```
# Retrait de ces outliers
train_corb = train_cor[-c(720,819,743,585),]
# Retrait des points anormaux identifiés lors de l'analyse exploratoire
train_corb = subset(train_corb, GrLivArea < 4000)
train_corb = subset(train_corb, GarageArea < 1250)
train_corb = subset(train_corb, TotalBsmtSF < 4000)
train_corb = subset(train_corb, X1stFlrSF < 4500)
train_corb = subset(train_corb, MasVnrArea < 1000)

#nouveau modèle sans les outliers
model_corb = lm(log(SalePrice)~., data=train_corb)
```

###A8 Comparaison des AIC afin de choisir le modèle qui sera étudié plus en avant

```
AIC=c(extractAIC(model_cor)[2],extractAIC(model_corb)[2])
names(AIC)=c('modèle_cor_base','model_cor_sans_outliers1')
AIC
```

```
##          modèle_cor_base model_cor_sans_outliers1
##                -3932.455                -4362.170
```

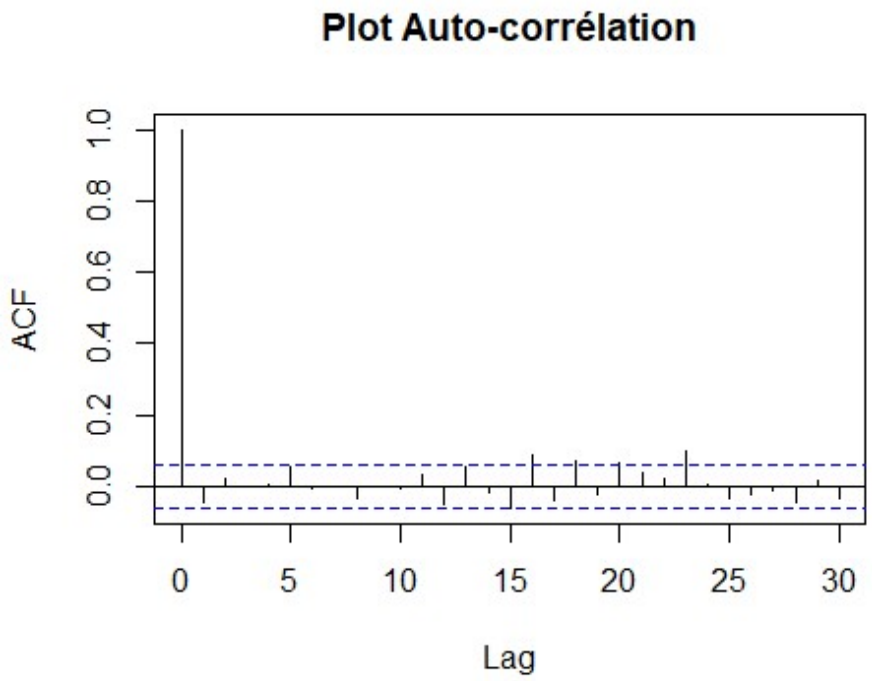
####A8 Analyse de validité du modèle retenu (modèle sur la base des variables sélectionnées par le corrplot)

####modèle de référence

```
ncvTest(model_cor)
```

```
## Non-constant Variance Score Test
## Variance formula: ~ fitted.values
## Chisquare = 593.7835, Df = 1, p = < 2.22e-16
```

```
acf(residuals(model_cor), main="Plot Auto-corrélation")
```




```
shapiro.test(residuals(model_cor))
```

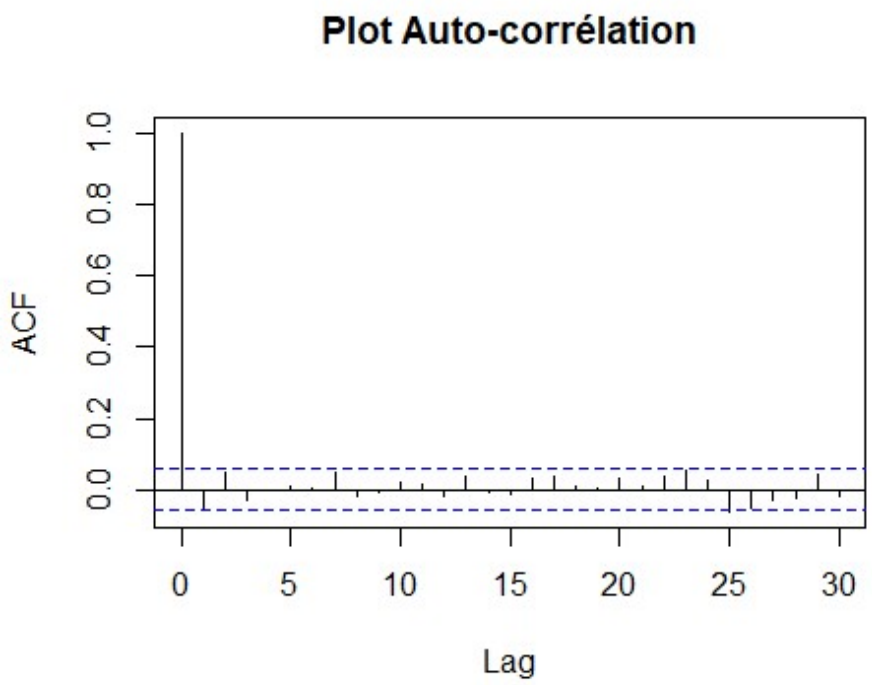
```
##
##  Shapiro-Wilk normality test
##
## data:  residuals(model_cor)
## W = 0.84297, p-value < 2.2e-16
```

####modèle sans la 1iere série d'outliers

```
ncvTest(model_corb)
```

```
## Non-constant Variance Score Test
## Variance formula: ~ fitted.values
## Chisquare = 23.12813, Df = 1, p = 1.5156e-06
```

```
acf(residuals(model_corb), main="Plot Auto-corrélation")
```




```
shapiro.test(residuals(model_corb))

##
##  Shapiro-Wilk normality test
##
## data:  residuals(model_corb)
## W = 0.96546, p-value = 2.243e-15
```

###Annexe 9 Etude de validé du modèle issu de selection des variables par le Random Forest

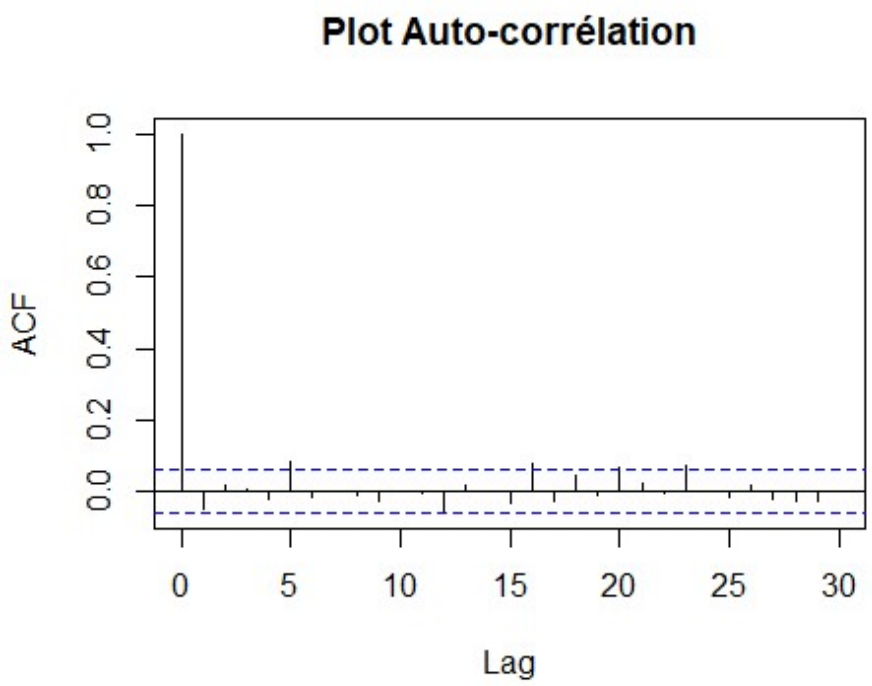
###A9 Liste des variables par ordre d'importance

```
#liste des variables par ordre d'importance décroissant
rf$importance[order(rf$importance[, 1], decreasing = TRUE), ]

##   OverallQual     GrLivArea    GarageCars   TotalBsmtSF     ExterQual
##  1.397968e+12  8.823408e+11  5.496264e+11  3.869813e+11  3.768270e+11
##     X1stFlrSF     YearBuilt    GarageArea    BsmtFinSF1      BsmtQual
##  3.232414e+11  2.910030e+11  2.856899e+11  1.843184e+11  1.185917e+11
##      FullBath       LotArea     X2ndFlrSF   KitchenQual  YearRemodAdd
##  9.857503e+10  9.267303e+10  8.757150e+10  7.633005e+10  7.608326e+10
##  TotRmsAbvGrd    MasVnrArea  GarageFinish    Fireplaces   GarageYrBlt
##  6.939656e+10  6.523113e+10  6.211332e+10  5.794426e+10  5.773619e+10
##   LotFrontage     BsmtUnfSF  Neighborhood   OverallCond   OpenPorchSF
##  4.593170e+10  3.676621e+10  2.795717e+10  2.779060e+10  2.708097e+10
##    WoodDeckSF  BsmtFinType1    GarageType        MoSold    MSSubClass
##  2.707806e+10  1.823832e+10  1.764499e+10  1.646332e+10  1.476076e+10
## SaleCondition      HalfBath      SaleType        YrSold  BsmtExposure
```

```
##  1.450268e+10  1.361127e+10  1.333110e+10  1.274482e+10  1.259491e+10
##  BedroomAbvGr    HouseStyle      MSZoning  BsmtFullBath     HeatingQC
##  1.214196e+10  1.159260e+10  1.078349e+10  1.055438e+10  8.390687e+09
##     LandSlope   LandContour    CentralAir    MasVnrType  KitchenAbvGr
##  8.218881e+09  7.947420e+09  7.422317e+09  7.359125e+09  7.322878e+09
##     RoofStyle    Foundation      LotShape EnclosedPorch   ScreenPorch
##  7.179736e+09  6.763483e+09  6.373940e+09  5.776980e+09  5.631387e+09
##     LotConfig      BldgType    Functional    BsmtFinSF2    Condition1
##  5.326126e+09  5.244539e+09  5.079585e+09  5.069712e+09  3.314609e+09
##      BsmtCond  BsmtFinType2     ExterCond    PavedDrive    X3SsnPorch
##  2.748235e+09  2.717657e+09  2.488935e+09  2.010920e+09  1.516850e+09
##       MiscVal    GarageQual  BsmtHalfBath    GarageCond  LowQualFinSF
##  1.194001e+09  1.075611e+09  1.035609e+09  5.853420e+08  5.695061e+08
##      PoolArea        Street     Utilities
##  5.630546e+08  3.688911e+08  1.341557e+07
```

####Analyse de validée du modèle de référence

```
ncvTest(model_rf)

## Non-constant Variance Score Test
## Variance formula: ~ fitted.values
## Chisquare = 153.0601, Df = 1, p = < 2.22e-16

acf(residuals(model_rf), main="Plot Auto-corrélation")
```




```
shapiro.test(residuals(model_rf))

##
##  Shapiro-Wilk normality test
##
```

```
## data:  residuals(model_rf)
## W = 0.8747, p-value < 2.2e-16
```

####A9 Visualisation des outliers issus du test de Bonferonni

```
#1ier retrait
outlierTest(model_rf)
```

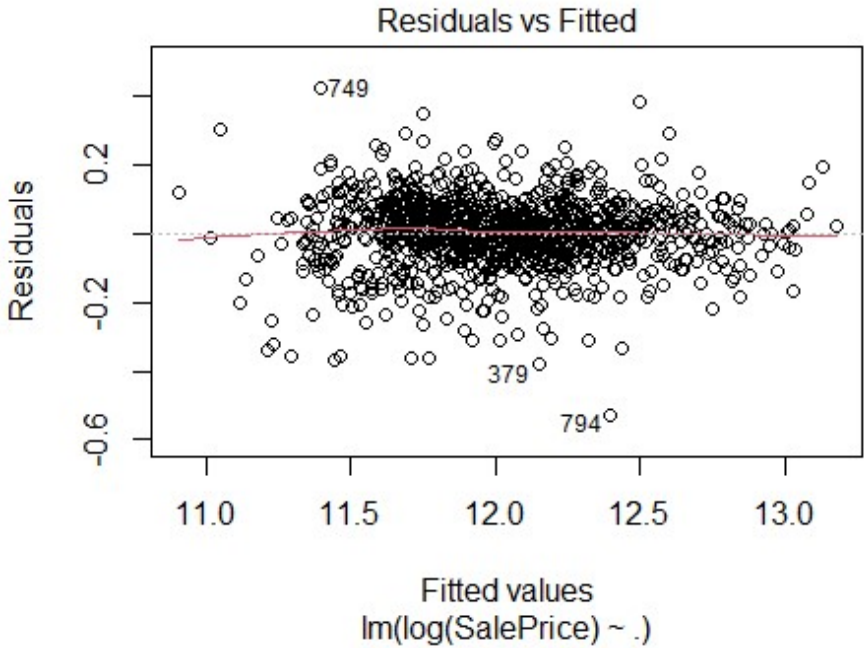
```
##        rstudent unadjusted p-value Bonferroni p
## 720  -16.084003         4.5401e-52   4.9669e-49
## 819   -9.240525         1.3790e-19   1.5087e-16
## 743   -5.484252         5.2307e-08   5.7224e-05
## 661   -4.809578         1.7392e-06   1.9027e-03
## 1079  -4.746666         2.3618e-06   2.5838e-03
## 814   -4.716912         2.7261e-06   2.9824e-03
## 404    4.108955         4.2920e-05   4.6955e-02
```

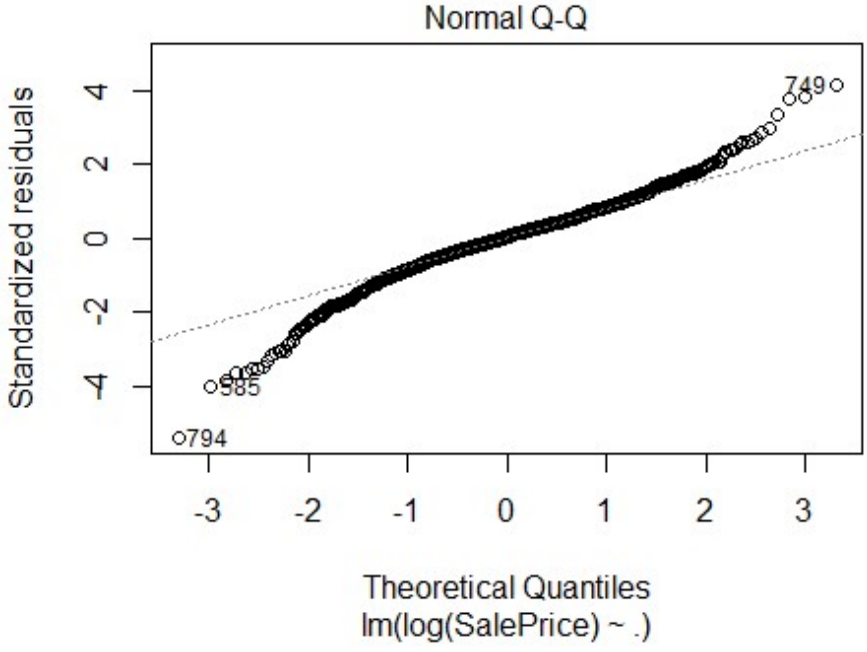
```
# 2ieme retrait des outliers
outlierTest(model_rfb)
```

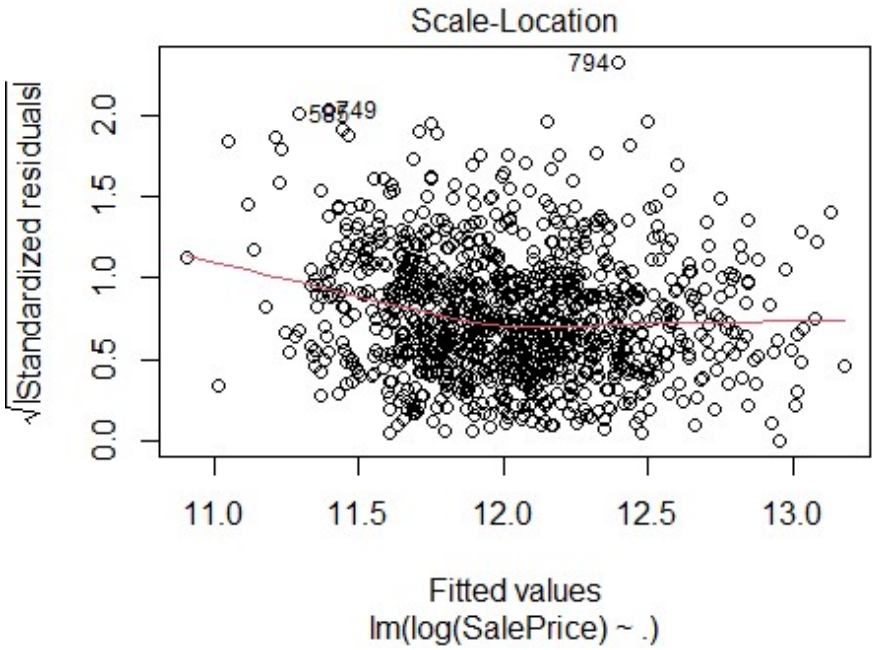
```
##      rstudent unadjusted p-value Bonferroni p
## 794 -5.488368         5.1287e-08   5.5441e-05
## 749  4.184823         3.1021e-05   3.3534e-02
```

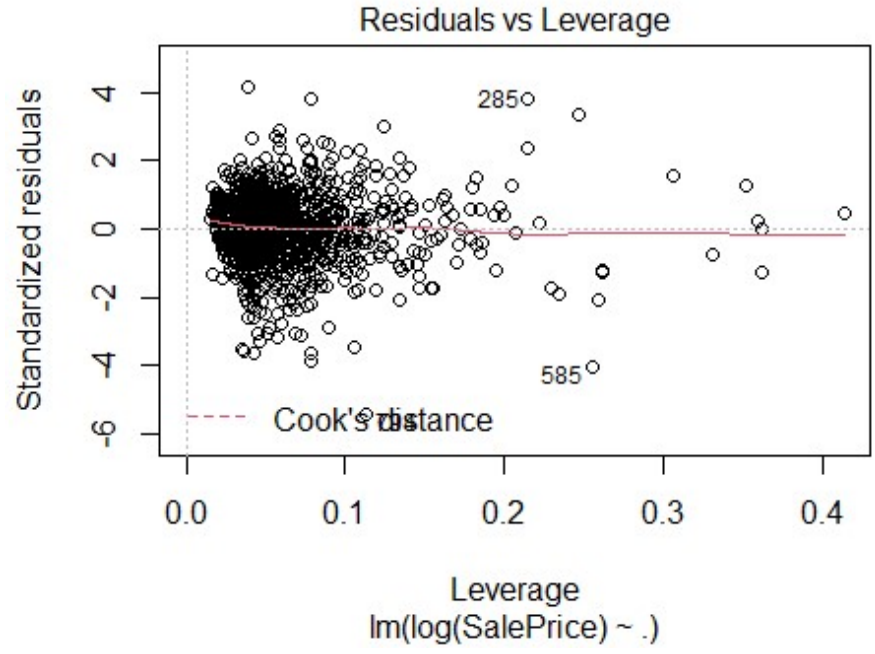
####A9 Analyse de validité du modèle issu de la sélection des variables par le RandomForest et sans les outliers

```
plot(model_rfb)
```

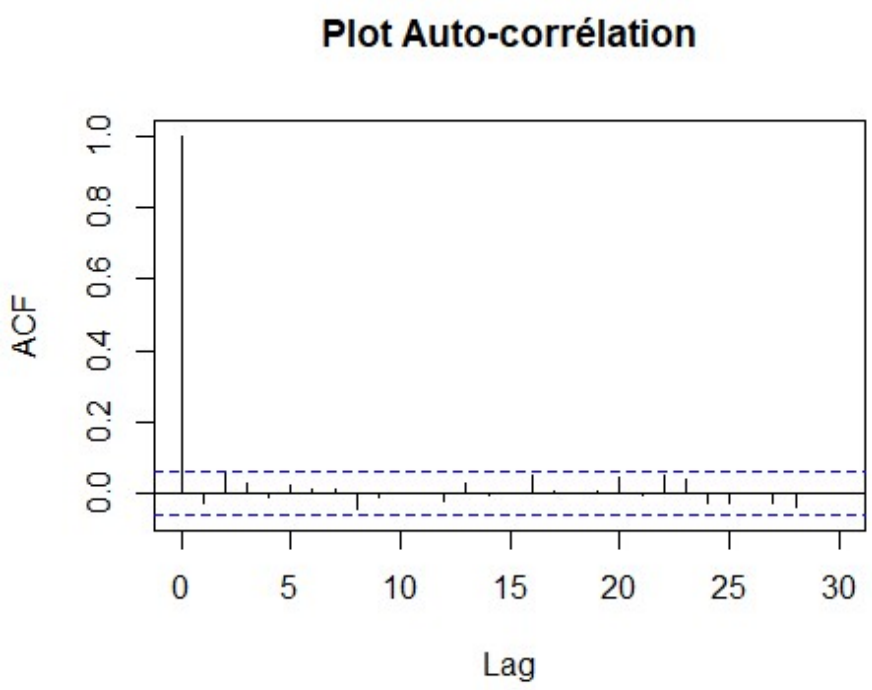

Residuals vs Fitted
749
379
794
Residuals
0.2
-0.2
-0.6
11.0
11.5
12.0
12.5
13.0
Fitted values
lm(log(SalePrice) ~ .)


Normal Q-Q
749
585
794
Standardized residuals
4
2
0
-2
-4
-3
-2
-1
0
1
2
3
Theoretical Quantiles
lm(log(SalePrice) ~ .)






```
ncvTest(model_rfb)

## Non-constant Variance Score Test
## Variance formula: ~ fitted.values
## Chisquare = 30.16634, Df = 1, p = 3.9653e-08

acf(residuals(model_rfb), main="Plot Auto-corrélation")
```




```
shapiro.test(residuals(model_rfb))
```

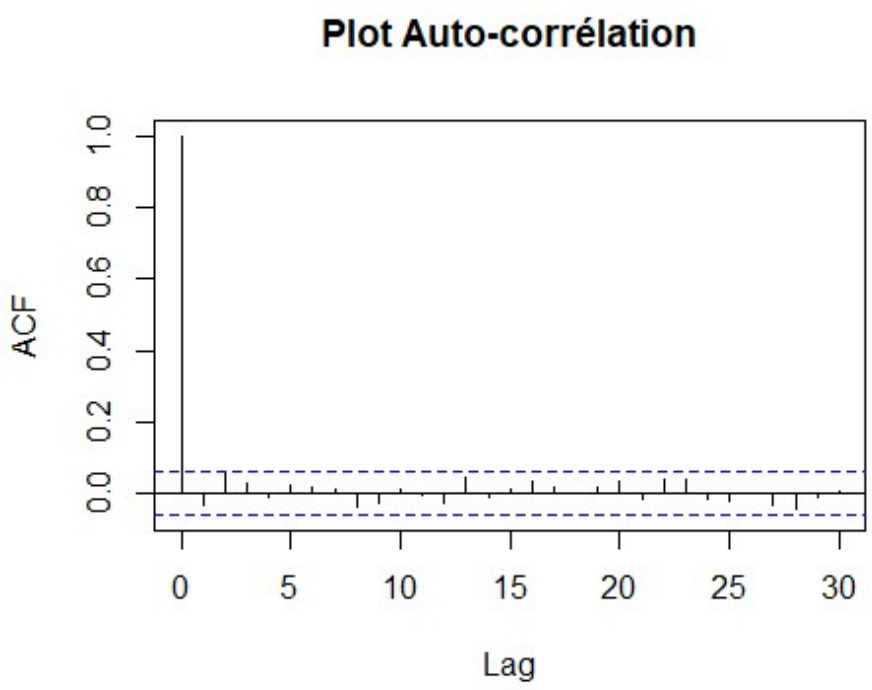
```
##
##  Shapiro-Wilk normality test
##
## data:  residuals(model_rfb)
## W = 0.96957, p-value = 2.772e-14
```

####Anayse du modèle sans la 2ieme série d'outliers

```
ncvTest(model_rfc)
```

```
## Non-constant Variance Score Test
## Variance formula: ~ fitted.values
## Chisquare = 30.45916, Df = 1, p = 3.4097e-08
```

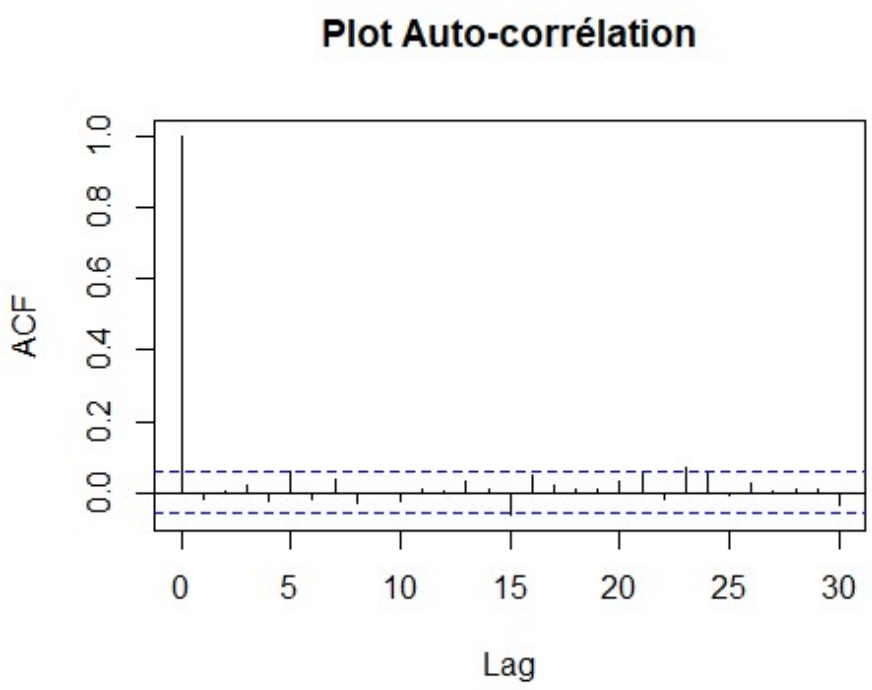
```
acf(residuals(model_rfc), main="Plot Auto-corrélation")
```




```
shapiro.test(residuals(model_rfc))

##
##  Shapiro-Wilk normality test
##
## data:  residuals(model_rfc)
## W = 0.96938, p-value = 2.547e-14
```

###Annexe 10 Etude de validé du modèle issu des transformation des variables "Neighborhood", "AGrLivArea" et "TotalBsmtSF"

```
ncvTest(model_var)

## Non-constant Variance Score Test
## Variance formula: ~ fitted.values
## Chisquare = 48.34336, Df = 1, p = 3.5775e-12

acf(residuals(model_var), main="Plot Auto-corrélation")
```




```
shapiro.test(residuals(model_var))
```

```
##
##  Shapiro-Wilk normality test
##
## data:  residuals(model_var)
## W = 0.8944, p-value < 2.2e-16
```

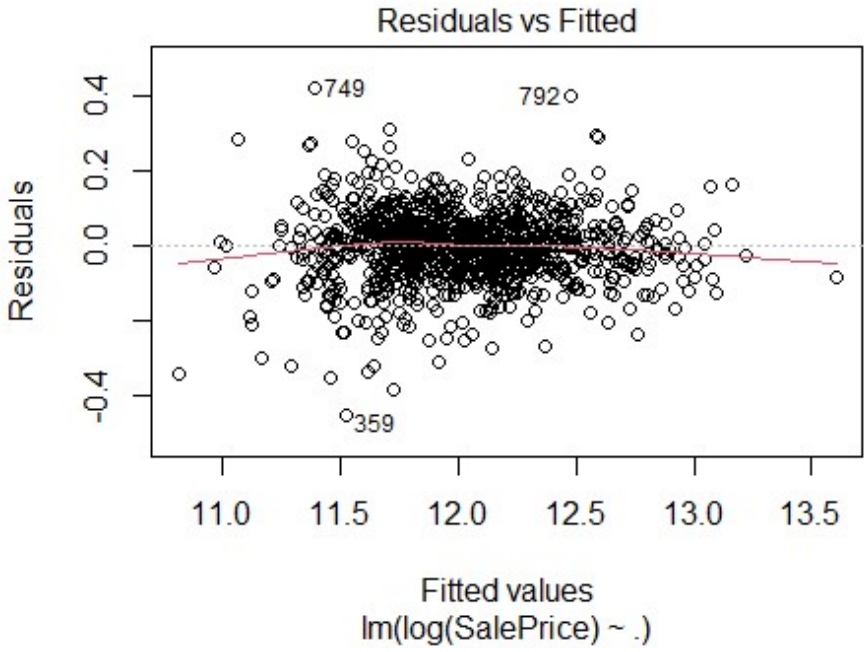
```
# visualisation des outliers issus du test de Bonferonni
outlierTest(model_var)
```

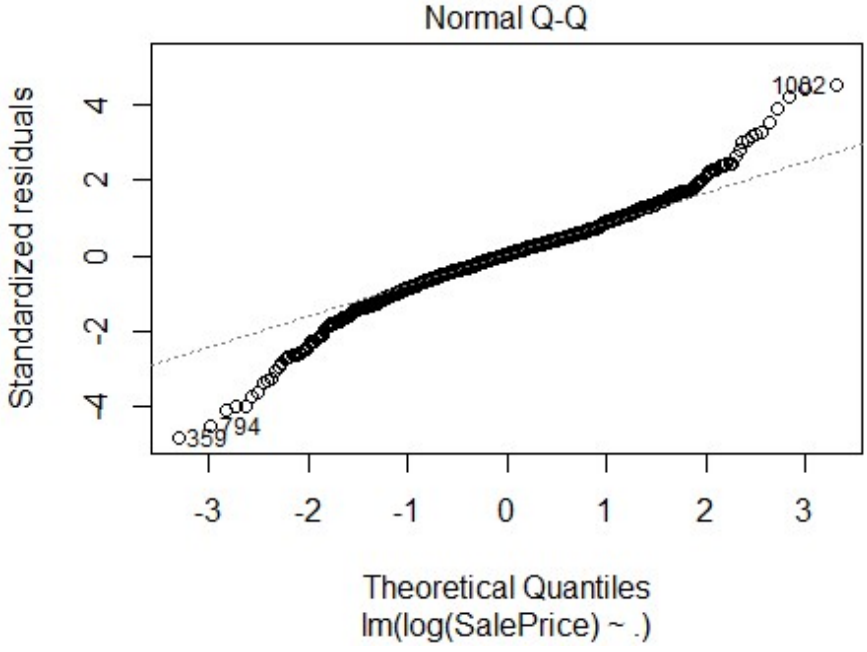
```
##        rstudent unadjusted p-value Bonferroni p
## 720  -13.984208         1.6612e-40   1.8057e-37
## 819  -10.012372         1.7219e-22   1.8717e-19
## 743   -5.904188         4.9479e-09   5.3784e-06
## 814   -5.135405         3.4230e-07   3.7208e-04
## 1079  -4.531641         6.6075e-06   7.1823e-03
```

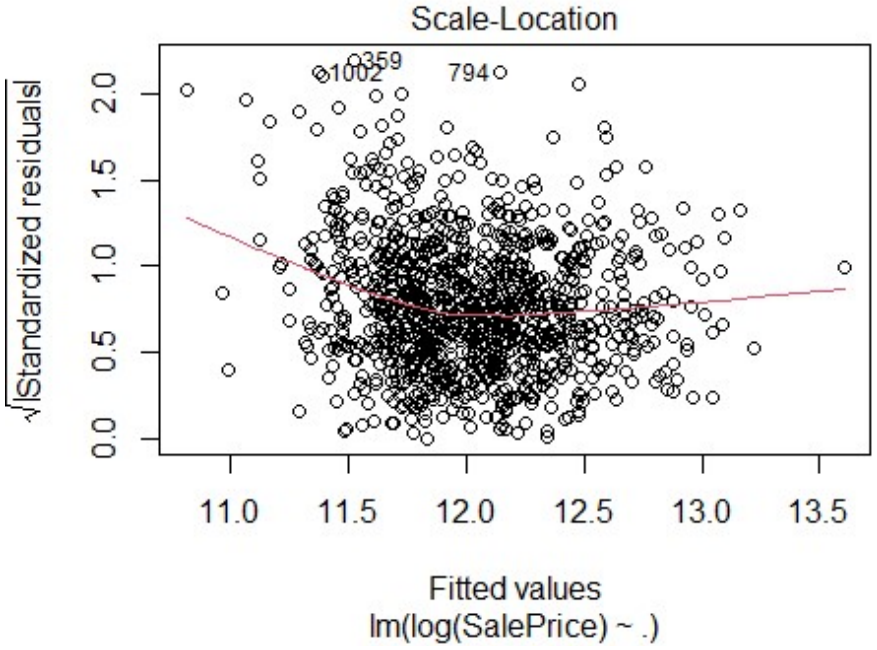
```
# visualisation des outliers résiduels (2ieme série)
outlierTest(model_varb)
```

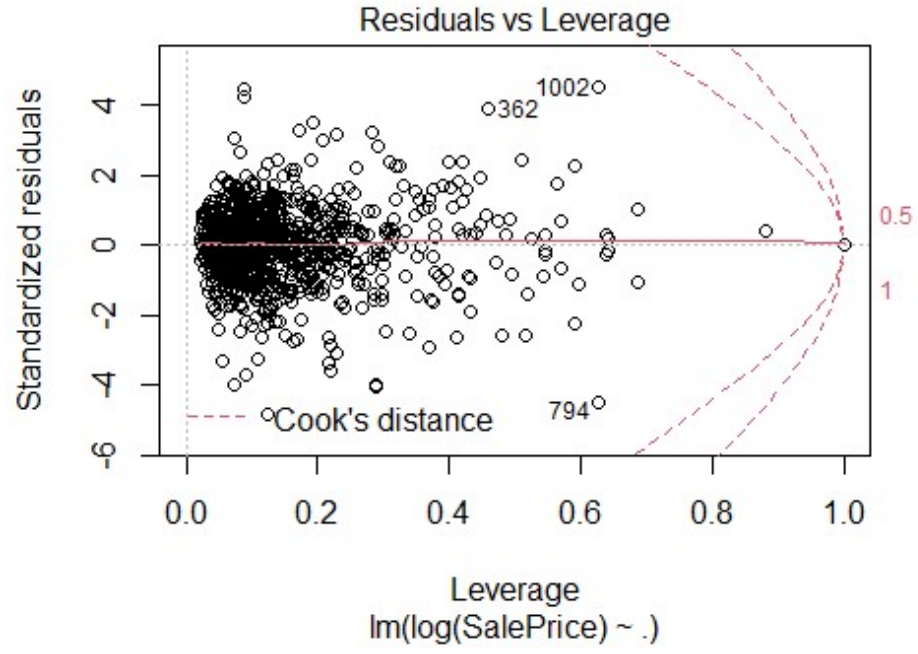
```
##       rstudent unadjusted p-value Bonferroni p
## 359  -4.881334         1.2399e-06    0.0013366
## 1002  4.555487         5.9226e-06    0.0063846
## 794  -4.555487         5.9226e-06    0.0063846
## 749   4.483872         8.2460e-06    0.0088892
## 792   4.272411         2.1329e-05    0.0229930
## 661  -4.108328         4.3370e-05    0.0467530
```

```
#Evaluation du modèle avec la 1iere série d'outliers retirée
plot(model_varb)
```

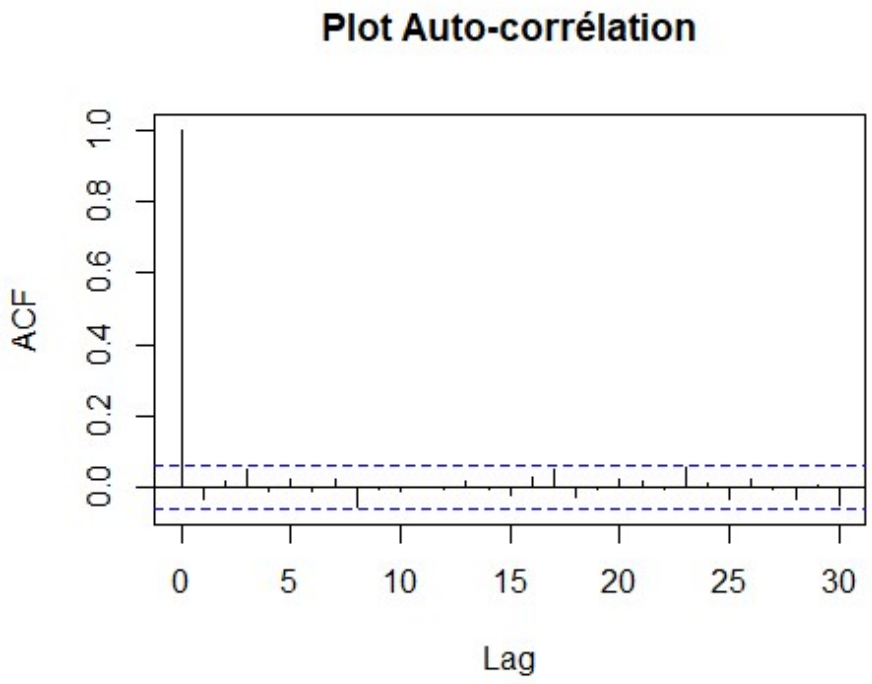

Residuals vs Fitted
Residuals
749
792
359
0.4
0.2
0.0
-0.4
11.0
11.5
12.0
12.5
13.0
13.5
Fitted values
lm(log(SalePrice) ~ .)


Normal Q-Q
Standardized residuals
1082
359
794
4
2
0
-2
-4
-3
-2
-1
0
1
2
3
Theoretical Quantiles
lm(log(SalePrice) ~ .)

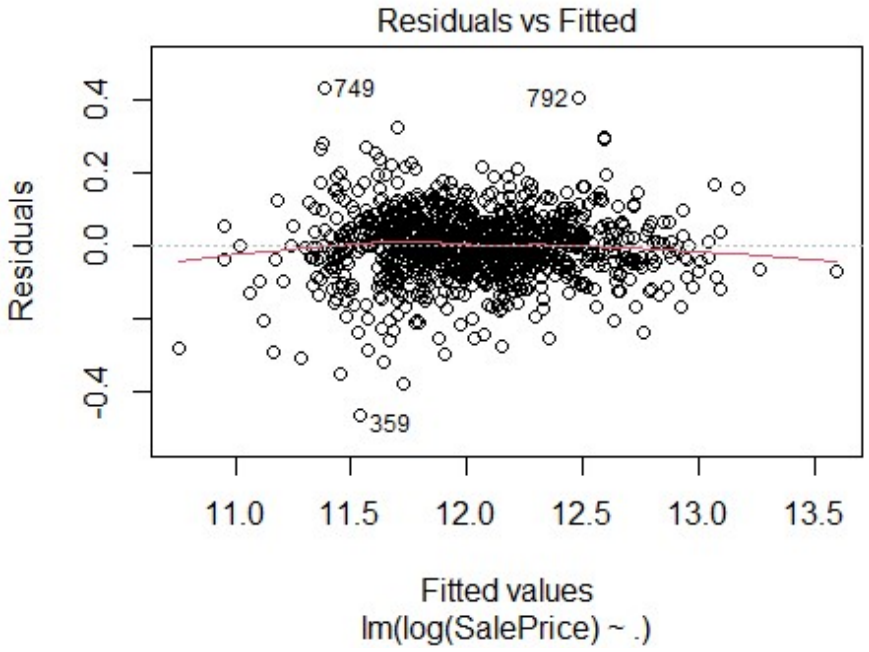


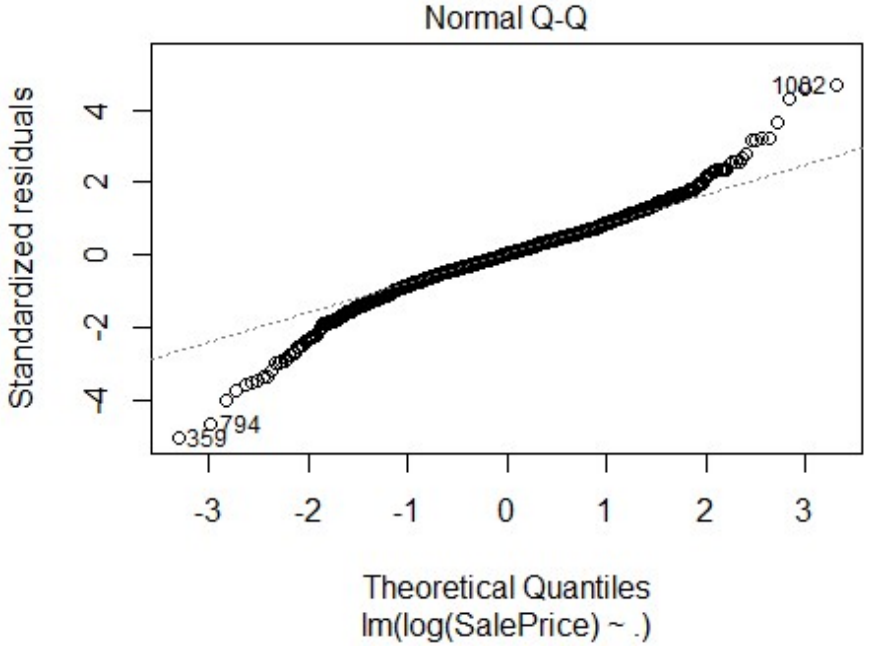


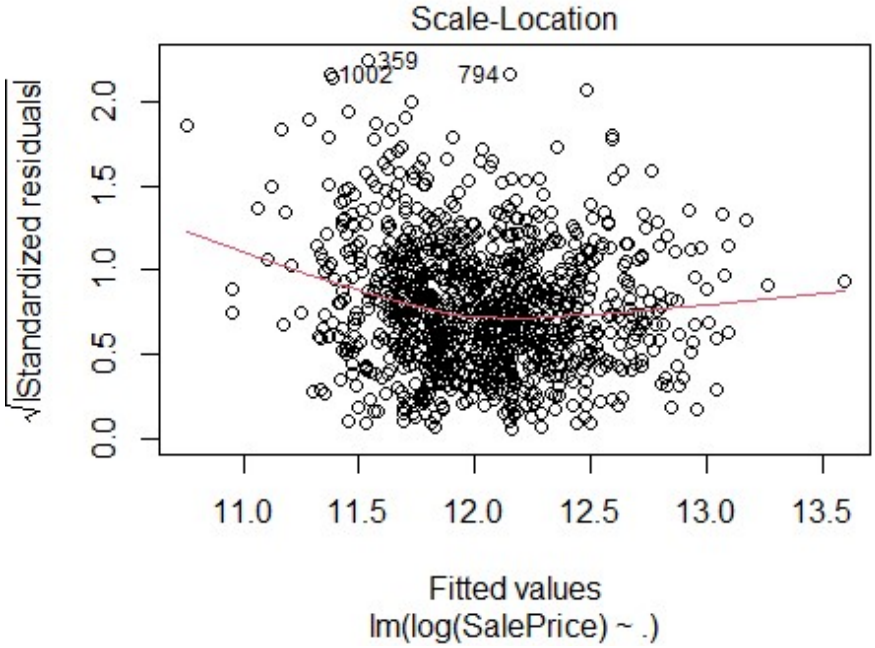

```
ncvTest(model_varb)

## Non-constant Variance Score Test
## Variance formula: ~ fitted.values
## Chisquare = 52.74501, Df = 1, p = 3.7979e-13

acf(residuals(model_varb), main="Plot Auto-corrélation")
```

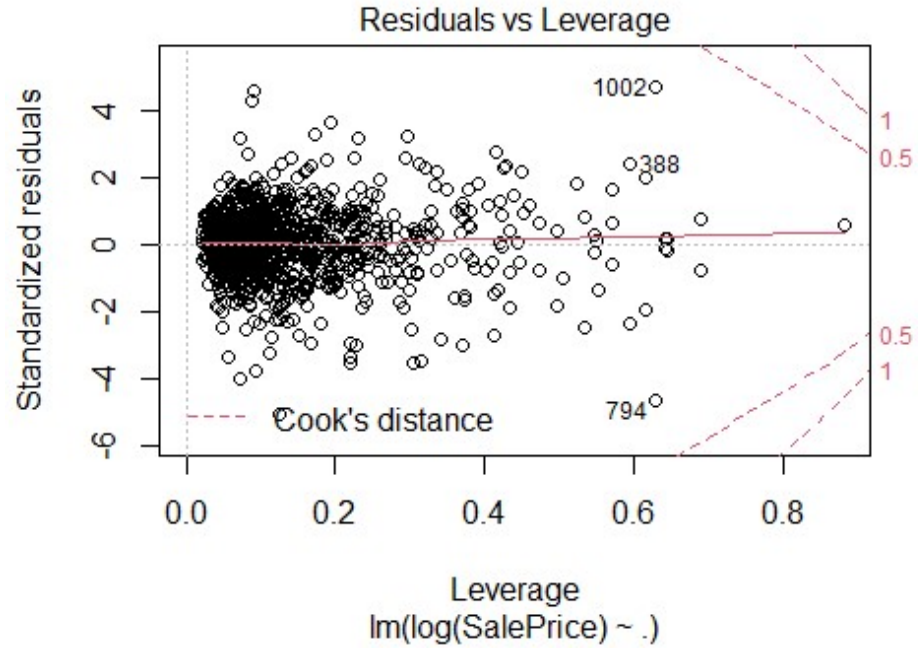



```
shapiro.test(residuals(model_varb))

##
##  Shapiro-Wilk normality test
##
## data:  residuals(model_varb)
## W = 0.97303, p-value = 2.488e-13

#Evaluation du modèle avec le retrait de la 2ieme sértie d'outliers
plot(model_varc)
```

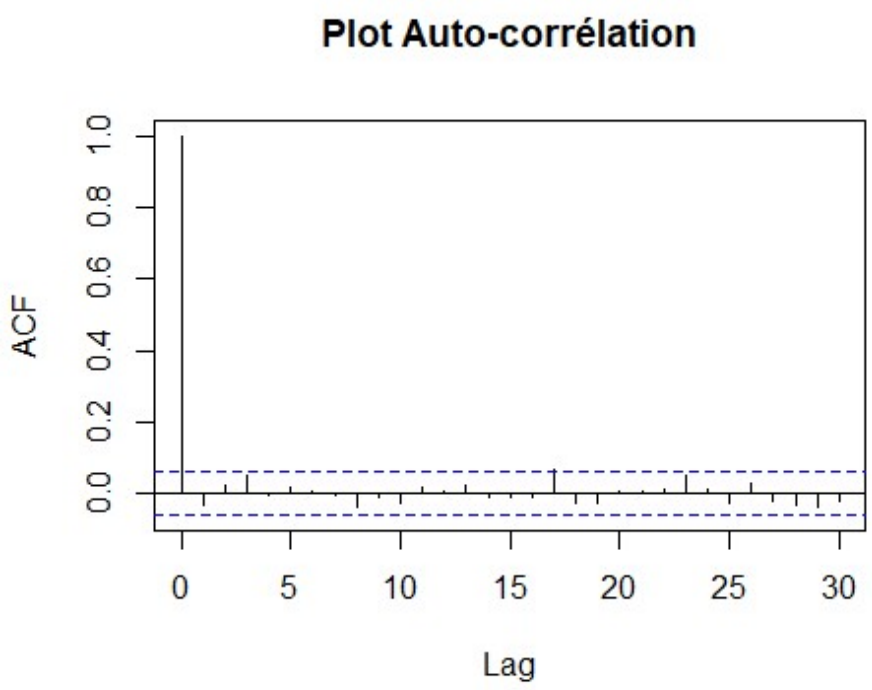

Residuals vs Fitted
Residuals
749
792
359
0.4
0.2
0.0
-0.4
11.0
11.5
12.0
12.5
13.0
13.5
Fitted values
lm(log(SalePrice) ~ .)


Normal Q-Q
Standardized residuals
1082
794
359
4
2
0
-2
-4
-3
-2
-1
0
1
2
3
Theoretical Quantiles
lm(log(SalePrice) ~ .)






```
ncvTest(model_varc)

## Non-constant Variance Score Test
## Variance formula: ~ fitted.values
## Chisquare = 44.38036, Df = 1, p = 2.7039e-11

acf(residuals(model_varc), main="Plot Auto-corrélation")
```




```
shapiro.test(residuals(model_varc))

##
##  Shapiro-Wilk normality test
##
## data:  residuals(model_varc)
## W = 0.97368, p-value = 4.299e-13
```

####Annexe 11 : Essai d'un modéle sans sélection, sans outliers mais normalisé

####A11 étape 1 : fusion des jeux de données train & test et préparation des données

```
train_an = subset(df_train, select=-c(Condition2, RoofMatl, Exterior1st, Exter
ior2nd, Heating, Electrical))
testt_an = subset(df_testt, select=-c(Condition2, RoofMatl, Exterior1st, Exter
ior2nd, Heating, Electrical))

# outliers issue de bonferonni
train_an = train_an[-c(720,819,743,661,814),]
testt_an = testt_an[-c(720,819,743,661,814),]

# outliers mis en évidence lors de l'analyse
train_an = subset(train_an, GrLivArea < 4000)
train_an = subset(train_an, GarageArea < 1250)
train_an = subset(train_an, TotalBsmtSF < 4000)
train_an = subset(train_an, MasVnrArea < 1000)
train_an = subset(train_an, TotalBsmtSF < 4000)
train_an = subset(train_an, X1stFlrSF < 4500)
```

```
all_tmp <- rbind(train_an, testt_an)
dim(all_tmp)
```

```
## [1] 1449   69
```

```
# récupération des index des variables numériques
numericVars <- which(sapply(all_tmp, is.numeric))
numericVarNames <- names(numericVars)
# affichage pour vérification
print(numericVarNames)
```

```
##  [1] "MSSubClass"    "LotFrontage"   "LotArea"       "OverallQual"
##  [5] "OverallCond"   "YearBuilt"     "YearRemodAdd"  "MasVnrArea"
##  [9] "BsmtFinSF1"    "BsmtFinSF2"    "BsmtUnfSF"     "TotalBsmtSF"
## [13] "X1stFlrSF"     "X2ndFlrSF"     "LowQualFinSF"  "GrLivArea"
## [17] "BsmtFullBath"  "BsmtHalfBath"  "FullBath"      "HalfBath"
## [21] "BedroomAbvGr"  "KitchenAbvGr"  "TotRmsAbvGrd"  "Fireplaces"
## [25] "GarageYrBlt"   "GarageCars"    "GarageArea"    "WoodDeckSF"
## [29] "OpenPorchSF"   "EnclosedPorch" "X3SsnPorch"    "ScreenPorch"
## [33] "PoolArea"      "MiscVal"       "MoSold"        "YrSold"
## [37] "SalePrice"
```

```
# récupération d'un dataframe avec uniquement les variables numériques
DFnumeric <- all_tmp[, names(all_tmp) %in% numericVarNames]
# on met de coté la target
DFtarget = subset(DFnumeric, select=c(SalePrice))
# on retire la target
DFnumeric = subset(DFnumeric, select=-c(SalePrice))
# récupération d'un df avec toutes les variables non numériques
DFfactors <- all_tmp[, !(names(all_tmp) %in% numericVarNames)]
# vérification des dimensions
dim(DFnumeric)
```

```
## [1] 1449   36
```

```
dim(DFtarget)
```

```
## [1] 1449    1
```

```
dim(DFfactors)
```

```
## [1] 1449   32
```

####A11 Etape 2 : Scaling

```
PreNum <- preProcess(DFnumeric, method=c("center", "scale"))
print(PreNum)
```

```
## Created from 1449 samples and 36 variables
##
## Pre-processing:
##   - centered (36)
##   - ignored (0)
##   - scaled (36)
```

```
DFnorm <- predict(PreNum, DFnumeric)
dim(DFnorm)

## [1] 1449   36
```

A noter qu'en procédant ainsi il y a une fuite de forme lors du scale, il aurait fallu faire un fit_transform sur le train et un simple transform sur le test, opération qui a échouée

```
# one hot encoding de toutes les variables qualitatives
DFdummies <- as.data.frame(model.matrix(~.-1, DFfactors))

# fusion de tous les dataframes avec les différents prédicateurs pour reformer
le jeu de donnée initial
all_tmp <- cbind(DFnorm, DFdummies, DFtarget)
dim(all_tmp)

## [1] 1449  182

# on split à nouveau le train et le test en conservant les dimensions
# prévues initialement
train_ = all_tmp[0:1084, ]
testt_ = all_tmp[1085:1449, ]
```

On voit que toutes les colonnes sont bien présentes: - catégoriques one hot encoded (avec des 0 ou 1), - les numériques scalées - et la target inchangée

```
head(train_[,c(33:40, 182)])

##      PoolArea     MiscVal     MoSold     YrSold MSZoningC (all) MSZoningFV
## 1 -0.05823586 -0.08799221  0.2521076 -1.3690763               0          0
## 2 -0.05823586 -0.08799221  0.9930893  0.1360766               0          0
## 3 -0.05823586 -0.08799221  0.2521076 -1.3690763               0          0
## 4 -0.05823586 -0.08799221  0.9930893  0.1360766               0          0
## 5 -0.05823586 -0.08799221  2.1045617 -0.6164998               0          0
## 6 -0.05823586 -0.08799221 -0.1183832 -1.3690763               0          0
##   MSZoningRH MSZoningRL SalePrice
## 1          0          1    180000
## 2          0          1    127500
## 3          0          1     84500
## 4          0          1    118000
## 5          0          1    179000
## 6          0          1    250000

dim(train_)

## [1] 1084  182

dim(testt_)

## [1] 365 182

train_b_ = subset(train_, select=-c(SalePrice))
testt_b_ = subset(testt_, select=-c(SalePrice))
dim(train_b_)

## [1] 1084  181
```

```
dim(testt_b_)

## [1] 365 181

model_scale = lm(log(train_$SalePrice)~., data=train_b_)
#summary(model_scale)
```

Mais les résultats sont inchangés par rapport au meme modèle non scalé...

```
y_train_pred = (predict(model_scale, newdata=train_b_))
y_testt_pred = (predict(model_scale, newdata=testt_b_))

RMSE_train = c(sqrt(mean((exp(y_train_pred)-train_$SalePrice)^2)))
RMSE_testt = c(sqrt(mean((exp(y_testt_pred)-testt_$SalePrice)^2)))

print("RMSE sur le dataset de train:"); print(RMSE_train, digits=5)

## [1] "RMSE sur le dataset de train:"

## [1] 16303

print("RMSE sur le dataset de test:"); print(RMSE_testt, digits=5)

## [1] "RMSE sur le dataset de test:"

## [1] 21690
```

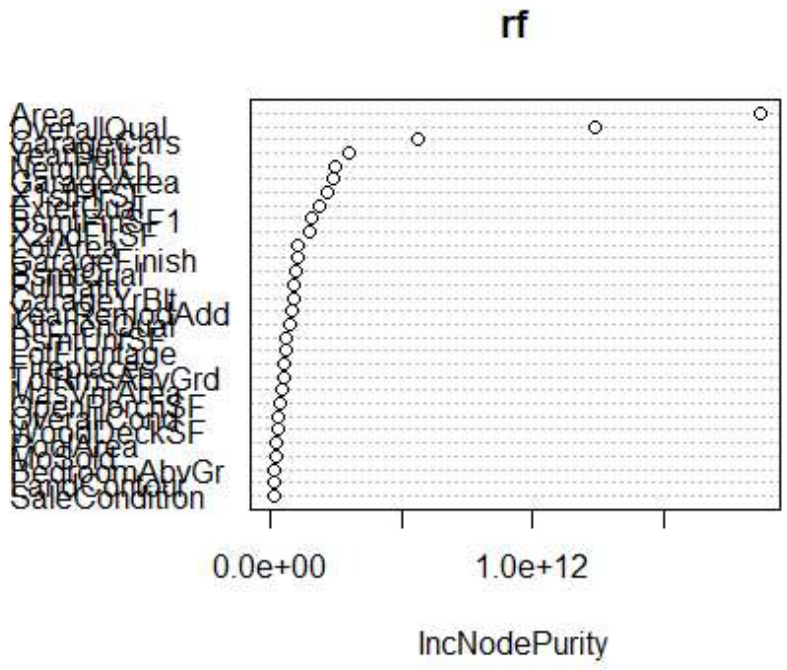
###Annexe 12 Modèle mixant modification de variables et sélection en utilisant les p-values de la synthèse du modèle de régression linéairessu d'une

```
# suppression des colonnes
train3 = subset(df_train, select=-c(Condition2, RoofMatl, Exterior1st, Exterio
r2nd, Heating, Electrical))
test3 = subset(df_testt, select=-c(Condition2, RoofMatl, Exterior1st, Exterior
2nd, Heating, Electrical))
```

###A12 Transformation des variables

```
train3$NeighRich[train3$Neighborhood %in% c('StoneBr', 'NridgHt', 'NoRidge')]
<- 2
test3$NeighRich[test3$Neighborhood %in% c('StoneBr', 'NridgHt', 'NoRidge')] <-
2

train3$NeighRich[!train3$Neighborhood %in% c('MeadowV', 'IDOTRR', 'BrDale', 'S
toneBr', 'NridgHt', 'NoRidge')] <- 1
test3$NeighRich[!test3$Neighborhood %in% c('MeadowV', 'IDOTRR', 'BrDale', 'Sto
neBr', 'NridgHt', 'NoRidge')] <- 1

train3$NeighRich[train3$Neighborhood %in% c('MeadowV', 'IDOTRR', 'BrDale')] <-
0
test3$NeighRich[test3$Neighborhood %in% c('MeadowV', 'IDOTRR', 'BrDale')] <- 0

train3 = (subset(train3, select=-c(Neighborhood)))
test3 = (subset(test3, select=-c(Neighborhood)))
```

```
# creation de la nouvelle variable
train3$Area <- train3$GrLivArea + train3$TotalBsmtSF
test3$Area <- test3$GrLivArea + test3$TotalBsmtSF

# suppression des anciennes
train3 = (subset(train3, select=-c(GrLivArea, TotalBsmtSF)))
test3 = (subset(test3, select=-c(GrLivArea, TotalBsmtSF)))

options(repr.plot.width = 10, repr.plot.height = 6)
rf = randomForest(train3$SalePrice~ .,data=train3)
varImpPlot(rf)
```



###A12 1iere étape de sélection des données sur la base du RandomForest

```
rf$importance[order(rf$importance[, 1], decreasing = TRUE), ]
```

```
##          Area   OverallQual    GarageCars     YearBuilt     NeighRich
##  1.866076e+12  1.239036e+12  5.618486e+11  2.970862e+11  2.494691e+11
##    GarageArea     X1stFlrSF     ExterQual    BsmtFinSF1     X2ndFlrSF
##  2.401265e+11  2.213078e+11  1.918120e+11  1.598391e+11  1.520710e+11
##       LotArea  GarageFinish      BsmtQual      FullBath   GarageYrBlt
##  1.053323e+11  1.033311e+11  9.613994e+10  9.155987e+10  9.143775e+10
##  YearRemodAdd   KitchenQual     BsmtUnfSF   LotFrontage    Fireplaces
##  8.638039e+10  7.602066e+10  6.059574e+10  6.018425e+10  5.760153e+10
##  TotRmsAbvGrd    MasVnrArea   OpenPorchSF   OverallCond    WoodDeckSF
##  5.092125e+10  4.539049e+10  3.884645e+10  3.449529e+10  2.810125e+10
##      PoolArea        MoSold  BedroomAbvGr   LandContour SaleCondition
##  2.714520e+10  2.232591e+10  1.936252e+10  1.713903e+10  1.598994e+10
##    GarageType  BsmtFinType1    MSSubClass  BsmtExposure        YrSold
##  1.520934e+10  1.493188e+10  1.456517e+10  1.444254e+10  1.283709e+10
```

```
##  KitchenAbvGr  BsmtFullBath      MSZoning     HeatingQC     RoofStyle
##  1.124268e+10  1.090292e+10  1.057858e+10  1.007918e+10  1.000207e+10
##      HalfBath    HouseStyle      SaleType     LandSlope      LotShape
##  9.561584e+09  9.540923e+09  9.303931e+09  9.254091e+09  9.003513e+09
##    CentralAir    Foundation    MasVnrType EnclosedPorch     LotConfig
##  8.900808e+09  8.335470e+09  7.848589e+09  7.257582e+09  6.666373e+09
##    Functional    Condition1   ScreenPorch      BldgType    BsmtFinSF2
##  5.927953e+09  5.837670e+09  4.733064e+09  4.538497e+09  4.237388e+09
##     ExterCond      BsmtCond  BsmtFinType2    X3SsnPorch    PavedDrive
##  3.732985e+09  3.208751e+09  2.959799e+09  2.604628e+09  2.503460e+09
##       MiscVal    GarageQual  BsmtHalfBath  LowQualFinSF    GarageCond
##  1.211203e+09  1.041269e+09  9.892773e+08  9.705400e+08  7.040070e+08
##        Street     Utilities
##  1.977126e+08  3.664566e+05
```

```
train3d = (subset(train3,select=c(Area,OverallQual,GarageCars,NeighRich,YearBu
ilt,ExterQual,X1stFlrSF,GarageArea,X2ndFlrSF,BsmtFinSF1,LotArea,FullBath,YearR
emodAdd,KitchenQual,GarageFinish,BsmtQual,GarageYrBlt,TotRmsAbvGrd,LotFrontage
,BsmtUnfSF,Fireplaces,MasVnrArea,OpenPorchSF,PoolArea,OverallCond,WoodDeckSF,M
oSold,BedroomAbvGr,SaleCondition,LandContour,GarageType,MSSubClass,BsmtExposur
e,BsmtFinType1,YrSold,KitchenAbvGr,BsmtFullBath,MSZoning,LotShape,HalfBath,Sal
eType,HouseStyle,RoofStyle,LandSlope,CentralAir,Foundation,MasVnrType,HeatingQ
C,LotConfig,EnclosedPorch,Condition1,BldgType,ScreenPorch,Functional,ExterCond
,BsmtFinSF2,BsmtFinType2,X3SsnPorch,BsmtCond,PavedDrive,LowQualFinSF,MiscVal,B
smtHalfBath,GarageQual,GarageCond,Street,Utilities,SalePrice)))
test3d = (subset(test3,select=c(Area,OverallQual,GarageCars,NeighRich,YearBuil
t,ExterQual,X1stFlrSF,GarageArea,X2ndFlrSF,BsmtFinSF1,LotArea,FullBath,YearRem
odAdd,KitchenQual,GarageFinish,BsmtQual,GarageYrBlt,TotRmsAbvGrd,LotFrontage,B
smtUnfSF,Fireplaces,MasVnrArea,OpenPorchSF,PoolArea,OverallCond,WoodDeckSF,MoS
old,BedroomAbvGr,SaleCondition,LandContour,GarageType,MSSubClass,BsmtExposure,
BsmtFinType1,YrSold,KitchenAbvGr,BsmtFullBath,MSZoning,LotShape,HalfBath,SaleT
ype,HouseStyle,RoofStyle,LandSlope,CentralAir,Foundation,MasVnrType,HeatingQC,
LotConfig,EnclosedPorch,Condition1,BldgType,ScreenPorch,Functional,ExterCond,B
smtFinSF2,BsmtFinType2,X3SsnPorch,BsmtCond,PavedDrive,LowQualFinSF,MiscVal,Bsm
tHalfBath,GarageQual,GarageCond,Street,Utilities,SalePrice)))

model_rf= lm(log(SalePrice)~., data=train3d)

#Utilisation de la synthèse pour identifier les variables faiblement corrélées
summary(model_rf)
```

```
##
## Call:
## lm(formula = log(SalePrice) ~ ., data = train3d)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.11528 -0.05527  0.00097  0.06660  0.41139
##
## Coefficients: (2 not defined because of singularities)
##                         Estimate Std. Error t value Pr(>|t|)
## (Intercept)           1.369e+01  6.644e+00   2.060 0.039651 *
## Area                  2.418e-04  1.070e-04   2.260 0.024023 *
## OverallQual           5.405e-02  6.357e-03   8.503  < 2e-16 ***
```

```
## GarageCars             6.875e-02  1.355e-02   5.074 4.70e-07 ***
## NeighRich              1.019e-01  1.533e-02   6.648 5.04e-11 ***
## YearBuilt              8.419e-04  4.492e-04   1.874 0.061231 .
## ExterQualFa            3.201e-02  6.932e-02   0.462 0.644326
## ExterQualGd            5.984e-02  3.046e-02   1.964 0.049788 *
## ExterQualTA            4.016e-02  3.365e-02   1.193 0.233046
## X1stFlrSF             -3.996e-05  1.092e-04  -0.366 0.714576
## GarageArea             4.585e-05  4.965e-05   0.923 0.356004
## X2ndFlrSF             -4.287e-05  1.032e-04  -0.416 0.677815
## BsmtFinSF1            -2.329e-04  1.110e-04  -2.099 0.036061 *
## LotArea                2.128e-06  6.757e-07   3.149 0.001691 **
## FullBath               3.399e-02  1.366e-02   2.488 0.013004 *
## YearRemodAdd           5.393e-04  3.557e-04   1.516 0.129846
## KitchenQualFa         -7.228e-02  3.878e-02  -1.864 0.062617 .
## KitchenQualGd         -7.748e-02  2.138e-02  -3.623 0.000306 ***
## KitchenQualTA         -9.022e-02  2.428e-02  -3.715 0.000215 ***
## GarageFinishRFn       -5.088e-03  1.226e-02  -0.415 0.678255
## GarageFinishUnf       -1.587e-02  1.538e-02  -1.032 0.302491
## BsmtQualFa            -1.216e-01  4.117e-02  -2.954 0.003210 **
## BsmtQualGd            -5.441e-02  2.052e-02  -2.652 0.008142 **
## BsmtQualTA            -8.204e-02  2.552e-02  -3.214 0.001352 **
## GarageYrBlt           -7.771e-04  4.131e-04  -1.881 0.060260 .
## TotRmsAbvGrd           1.209e-02  5.911e-03   2.046 0.041076 *
## LotFrontage           -1.124e-03  2.777e-04  -4.048 5.60e-05 ***
## BsmtUnfSF             -2.227e-04  1.098e-04  -2.027 0.042951 *
## Fireplaces             3.774e-02  8.381e-03   4.503 7.54e-06 ***
## MasVnrArea            -2.341e-05  3.724e-05  -0.629 0.529771
## OpenPorchSF            1.454e-05  7.383e-05   0.197 0.843937
## PoolArea              -2.047e-04  1.030e-04  -1.987 0.047193 *
## OverallCond            3.487e-02  5.595e-03   6.233 6.91e-10 ***
## WoodDeckSF             1.251e-04  3.935e-05   3.179 0.001524 **
## MoSold                 3.514e-05  1.583e-03   0.022 0.982296
## BedroomAbvGr           1.188e-02  8.559e-03   1.388 0.165375
## SaleConditionAdjLand  -7.571e-02  9.067e-02  -0.835 0.403933
## SaleConditionAlloca    9.292e-02  5.849e-02   1.589 0.112506
## SaleConditionFamily    1.143e-02  4.169e-02   0.274 0.783998
## SaleConditionNormal    5.542e-02  1.801e-02   3.076 0.002156 **
## SaleConditionPartial  -1.048e-01  1.049e-01  -0.999 0.318040
## LandContourHLS         1.077e-01  3.212e-02   3.352 0.000836 ***
## LandContourLow         9.949e-02  4.187e-02   2.376 0.017699 *
## LandContourLvl         8.722e-02  2.388e-02   3.653 0.000274 ***
## GarageTypeAttchd       8.982e-02  1.068e-01   0.841 0.400618
## GarageTypeBasment      5.889e-02  1.123e-01   0.524 0.600096
## GarageTypeBuiltIn      5.690e-02  1.088e-01   0.523 0.601003
## GarageTypeCarPort      5.211e-02  1.201e-01   0.434 0.664592
## GarageTypeDetchd       8.389e-02  1.065e-01   0.788 0.431049
## MSSubClass            -7.416e-04  5.075e-04  -1.461 0.144221
## BsmtExposureGd         2.273e-02  1.921e-02   1.183 0.237026
## BsmtExposureMn        -2.028e-02  1.910e-02  -1.062 0.288709
## BsmtExposureNo        -2.167e-02  1.362e-02  -1.591 0.111839
## BsmtFinType1BLQ       -2.419e-02  1.751e-02  -1.382 0.167430
## BsmtFinType1GLQ       -2.538e-03  1.620e-02  -0.157 0.875530
## BsmtFinType1LwQ       -4.664e-02  2.480e-02  -1.881 0.060324 .
```

```
## BsmtFinType1Rec      -2.365e-02  1.950e-02  -1.213 0.225460
## BsmtFinType1Unf      -6.639e-02  1.827e-02  -3.634 0.000294 ***
## YrSold               -2.307e-03  3.277e-03  -0.704 0.481633
## KitchenAbvGr         -6.068e-02  3.798e-02  -1.598 0.110458
## BsmtFullBath          3.796e-02  1.259e-02   3.014 0.002648 **
## MSZoningFV            2.950e-01  7.034e-02   4.194 3.00e-05 ***
## MSZoningRH            2.322e-01  7.780e-02   2.985 0.002910 **
## MSZoningRL            2.213e-01  6.720e-02   3.294 0.001026 **
## MSZoningRM            1.790e-01  6.603e-02   2.711 0.006827 **
## LotShapeIR2           3.079e-02  2.774e-02   1.110 0.267218
## LotShapeIR3          -1.703e-01  5.822e-02  -2.925 0.003530 **
## LotShapeReg          -5.411e-04  1.013e-02  -0.053 0.957394
## HalfBath              4.153e-02  1.342e-02   3.095 0.002023 **
## SaleTypeCon           1.433e-01  9.825e-02   1.459 0.144890
## SaleTypeConLD         1.262e-01  5.894e-02   2.141 0.032491 *
## SaleTypeConLI        -4.371e-02  8.529e-02  -0.512 0.608452
## SaleTypeConLw        -3.117e-02  8.954e-02  -0.348 0.727840
## SaleTypeCWD           9.883e-02  7.456e-02   1.325 0.185330
## SaleTypeNew           2.007e-01  1.085e-01   1.850 0.064596 .
## SaleTypeOth           5.834e-02  8.264e-02   0.706 0.480409
## SaleTypeWD           -4.950e-03  2.599e-02  -0.190 0.848989
## HouseStyle1.5Unf      2.570e-02  4.732e-02   0.543 0.587143
## HouseStyle1Story     -3.734e-03  2.672e-02  -0.140 0.888886
## HouseStyle2.5Fin     -1.215e-01  8.329e-02  -1.459 0.144952
## HouseStyle2.5Unf      3.480e-02  5.958e-02   0.584 0.559266
## HouseStyle2Story     -4.425e-02  2.201e-02  -2.011 0.044662 *
## HouseStyleSFoyer     -5.559e-03  3.861e-02  -0.144 0.885544
## HouseStyleSLvl        5.454e-03  3.344e-02   0.163 0.870463
## RoofStyleGable       -7.295e-02  6.386e-02  -1.142 0.253641
## RoofStyleGambrel     -3.228e-03  8.075e-02  -0.040 0.968124
## RoofStyleHip         -6.331e-02  6.464e-02  -0.979 0.327656
## RoofStyleMansard     -5.808e-02  8.261e-02  -0.703 0.482202
## RoofStyleShed         1.653e-01  1.580e-01   1.046 0.295639
## LandSlopeMod          6.326e-02  2.574e-02   2.458 0.014168 *
## LandSlopeSev         -4.605e-02  7.480e-02  -0.616 0.538299
## CentralAirY           5.276e-02  2.286e-02   2.308 0.021204 *
## FoundationCBlock      4.239e-02  1.949e-02   2.175 0.029851 *
## FoundationPConc       5.840e-02  2.190e-02   2.667 0.007786 **
## FoundationSlab       -4.154e-02  4.816e-02  -0.863 0.388592
## FoundationStone       8.299e-02  6.315e-02   1.314 0.189128
## FoundationWood        1.053e-02  1.018e-01   0.103 0.917597
## MasVnrTypeBrkFace     5.063e-02  4.536e-02   1.116 0.264583
## MasVnrTypeNone        5.168e-02  4.561e-02   1.133 0.257499
## MasVnrTypeStone       6.013e-02  4.750e-02   1.266 0.205897
## HeatingQCFa          -3.900e-02  2.764e-02  -1.411 0.158601
## HeatingQCGd          -2.937e-02  1.316e-02  -2.232 0.025841 *
## HeatingQCPo          -1.477e-01  1.537e-01  -0.961 0.337039
## HeatingQCTA          -3.691e-02  1.313e-02  -2.810 0.005051 **
## LotConfigCulDSac      2.576e-02  2.187e-02   1.178 0.239102
## LotConfigFR2         -4.868e-02  2.541e-02  -1.916 0.055719 .
## LotConfigFR3         -7.036e-02  7.864e-02  -0.895 0.371180
## LotConfigInside      -8.481e-03  1.181e-02  -0.718 0.472868
## EnclosedPorch         2.021e-04  7.876e-05   2.566 0.010428 *
```

```
## Condition1Feedr       6.212e-02  3.129e-02   1.985 0.047416 *
## Condition1Norm        1.316e-01  2.652e-02   4.962 8.30e-07 ***
## Condition1PosA        1.323e-01  5.744e-02   2.303 0.021513 *
## Condition1PosN        5.114e-02  4.412e-02   1.159 0.246738
## Condition1RRAe        2.318e-02  6.020e-02   0.385 0.700292
## Condition1RRAn        1.549e-01  4.238e-02   3.655 0.000272 ***
## Condition1RRNe        1.363e-01  1.345e-01   1.013 0.311221
## Condition1RRNn        3.115e-01  1.130e-01   2.757 0.005955 **
## BldgType2fmCon        4.329e-02  7.625e-02   0.568 0.570354
## BldgTypeDuplex        4.866e-02  4.836e-02   1.006 0.314597
## BldgTypeTwnhs        -4.848e-02  5.950e-02  -0.815 0.415438
## BldgTypeTwnhsE        1.729e-03  5.426e-02   0.032 0.974588
## ScreenPorch           2.704e-04  7.824e-05   3.456 0.000572 ***
## FunctionalMaj2       -1.663e-01  9.430e-02  -1.763 0.078198 .
## FunctionalMin1       -1.677e-03  5.790e-02  -0.029 0.976896
## FunctionalMin2       -6.757e-03  5.670e-02  -0.119 0.905169
## FunctionalMod        -6.905e-02  6.661e-02  -1.037 0.300215
## FunctionalSev        -3.027e-01  1.760e-01  -1.720 0.085732 .
## FunctionalTyp         3.858e-02  4.931e-02   0.782 0.434202
## ExterCondFa           7.603e-02  1.047e-01   0.726 0.467805
## ExterCondGd           5.747e-02  9.858e-02   0.583 0.560095
## ExterCondPo          -4.291e-02  1.776e-01  -0.242 0.809198
## ExterCondTA           6.488e-02  9.841e-02   0.659 0.509905
## BsmtFinSF2           -1.412e-04  1.193e-04  -1.184 0.236786
## BsmtFinType2BLQ      -3.685e-02  5.153e-02  -0.715 0.474716
## BsmtFinType2GLQ      -1.303e-02  6.060e-02  -0.215 0.829779
## BsmtFinType2LwQ       1.304e-02  4.971e-02   0.262 0.793118
## BsmtFinType2Rec       1.297e-02  4.836e-02   0.268 0.788656
## BsmtFinType2Unf       4.631e-02  5.198e-02   0.891 0.373189
## X3SsnPorch            4.399e-04  1.539e-04   2.858 0.004362 **
## BsmtCondGd            2.311e-02  3.494e-02   0.661 0.508625
## BsmtCondPo           -2.542e-02  1.774e-01  -0.143 0.886091
## BsmtCondTA            1.161e-02  2.763e-02   0.420 0.674502
## PavedDriveP           1.429e-02  3.684e-02   0.388 0.698196
## PavedDriveY           2.325e-02  2.167e-02   1.073 0.283507
## LowQualFinSF                 NA         NA      NA       NA
## MiscVal               3.703e-06  8.198e-06   0.452 0.651597
## BsmtHalfBath          1.299e-02  2.120e-02   0.613 0.540168
## GarageQualFa         -1.959e-01  1.439e-01  -1.361 0.173935
## GarageQualGd         -8.481e-02  1.472e-01  -0.576 0.564549
## GarageQualPo         -2.273e-01  1.908e-01  -1.191 0.233975
## GarageQualTA         -1.536e-01  1.404e-01  -1.094 0.274367
## GarageCondFa         -2.725e-02  3.844e-02  -0.709 0.478613
## GarageCondGd         -2.269e-02  5.830e-02  -0.389 0.697172
## GarageCondPo          6.289e-02  8.102e-02   0.776 0.437754
## GarageCondTA                 NA         NA      NA       NA
## StreetPave            1.294e-01  7.383e-02   1.752 0.080080 .
## UtilitiesNoSeWa      -1.905e-01  1.489e-01  -1.280 0.201010
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.1297 on 940 degrees of freedom
```

```
## Multiple R-squared:  0.9065, Adjusted R-squared:  0.8911
## F-statistic: 59.16 on 154 and 940 DF,  p-value: < 2.2e-16

#suppression des variables faiblement corrélées
train3e=train3d[,-c(1,5,6,7,8,9,10,12,13,15,17,18,20,22,23,24,27,28,31,32,33,3
5,36,41,42,43,44,45,47,49,50,52,54,55,56,58,59,60,61,62,63,64,65)]

#modèle final
model_rfe= lm(log(train3e$SalePrice)~., data=train3e)

#analyse du modèle final
summary(model_rfe)

##
## Call:
## lm(formula = log(train3e$SalePrice) ~ ., data = train3e)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.04782 -0.08606  0.00097  0.08582  0.50074
##
## Coefficients:
##                          Estimate Std. Error t value Pr(>|t|)
## (Intercept)           1.041e+01  1.246e-01  83.536  < 2e-16 ***
## OverallQual           9.585e-02  6.398e-03  14.981  < 2e-16 ***
## GarageCars            9.341e-02  9.062e-03  10.307  < 2e-16 ***
## NeighRich             1.099e-01  1.702e-02   6.459 1.62e-10 ***
## LotArea               3.430e-06  5.680e-07   6.039 2.16e-09 ***
## KitchenQualFa        -1.383e-01  4.199e-02  -3.294 0.001019 **
## KitchenQualGd        -4.630e-02  2.335e-02  -1.983 0.047626 *
## KitchenQualTA        -1.035e-01  2.628e-02  -3.938 8.75e-05 ***
## BsmtQualFa           -1.557e-01  4.405e-02  -3.536 0.000424 ***
## BsmtQualGd           -6.271e-02  2.287e-02  -2.742 0.006210 **
## BsmtQualTA           -8.934e-02  2.765e-02  -3.230 0.001275 **
## LotFrontage           8.431e-04  2.508e-04   3.361 0.000805 ***
## Fireplaces            7.024e-02  8.821e-03   7.962 4.43e-15 ***
## OverallCond           3.111e-02  5.031e-03   6.183 9.06e-10 ***
## WoodDeckSF            1.904e-04  4.405e-05   4.323 1.69e-05 ***
## SaleConditionAdjLand  4.568e-02  9.392e-02   0.486 0.626806
## SaleConditionAlloca   1.401e-01  6.219e-02   2.253 0.024475 *
## SaleConditionFamily   4.246e-02  4.782e-02   0.888 0.374853
## SaleConditionNormal   5.015e-02  1.931e-02   2.597 0.009550 **
## SaleConditionPartial  9.995e-02  2.670e-02   3.743 0.000192 ***
## LandContourHLS        7.837e-02  3.599e-02   2.178 0.029643 *
## LandContourLow        8.604e-02  4.335e-02   1.985 0.047439 *
## LandContourLvl        6.047e-02  2.604e-02   2.322 0.020417 *
## BsmtFinType1BLQ      -1.765e-02  1.996e-02  -0.884 0.376773
## BsmtFinType1GLQ       3.574e-04  1.849e-02   0.019 0.984578
## BsmtFinType1LwQ      -3.887e-03  2.734e-02  -0.142 0.886950
## BsmtFinType1Rec       7.087e-03  2.173e-02   0.326 0.744349
## BsmtFinType1Unf      -5.404e-02  1.834e-02  -2.947 0.003279 **
## BsmtFullBath          3.170e-02  1.207e-02   2.627 0.008743 **
## MSZoningFV            1.994e-01  7.335e-02   2.719 0.006656 **
## MSZoningRH            1.791e-01  8.233e-02   2.175 0.029824 *
```

```
## MSZoningRL             1.733e-01  6.932e-02   2.500 0.012579 *
## MSZoningRM             9.433e-02  6.935e-02   1.360 0.174033
## LotShapeIR2            1.089e-02  3.181e-02   0.342 0.732195
## LotShapeIR3           -1.846e-01  6.571e-02  -2.810 0.005055 **
## LotShapeReg           -8.124e-03  1.112e-02  -0.730 0.465391
## HalfBath               5.445e-02  1.048e-02   5.195 2.47e-07 ***
## FoundationCBlock       2.545e-02  1.878e-02   1.356 0.175543
## FoundationPConc        5.595e-02  2.192e-02   2.552 0.010841 *
## FoundationSlab        -5.210e-02  4.591e-02  -1.135 0.256770
## FoundationStone        1.531e-01  6.757e-02   2.266 0.023662 *
## FoundationWood         6.174e-02  1.168e-01   0.529 0.597200
## HeatingQCFa           -9.481e-02  2.940e-02  -3.225 0.001298 **
## HeatingQCGd           -4.102e-02  1.481e-02  -2.770 0.005713 **
## HeatingQCPo           -1.308e-01  1.697e-01  -0.771 0.441071
## HeatingQCTA           -4.458e-02  1.422e-02  -3.136 0.001762 **
## Condition1Feedr        4.348e-02  3.533e-02   1.231 0.218768
## Condition1Norm         1.053e-01  2.944e-02   3.576 0.000365 ***
## Condition1PosA         1.525e-01  6.422e-02   2.375 0.017710 *
## Condition1PosN         1.264e-01  4.913e-02   2.572 0.010248 *
## Condition1RRAe         1.406e-03  6.662e-02   0.021 0.983171
## Condition1RRAn         1.498e-01  4.833e-02   3.100 0.001985 **
## Condition1RRNe         1.233e-01  1.606e-01   0.767 0.442991
## Condition1RRNn         2.497e-01  1.259e-01   1.983 0.047625 *
## ScreenPorch            3.024e-04  8.929e-05   3.387 0.000733 ***
## BsmtFinType2BLQ       -9.752e-02  5.531e-02  -1.763 0.078200 .
## BsmtFinType2GLQ       -3.601e-02  6.849e-02  -0.526 0.599131
## BsmtFinType2LwQ       -2.124e-02  5.294e-02  -0.401 0.688266
## BsmtFinType2Rec       -3.884e-02  5.224e-02  -0.743 0.457372
## BsmtFinType2Unf       -3.668e-02  4.622e-02  -0.794 0.427583
## StreetPave             2.108e-01  7.528e-02   2.800 0.005203 **
## UtilitiesNoSeWa       -2.498e-01  1.623e-01  -1.539 0.124134
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.1564 on 1033 degrees of freedom
## Multiple R-squared:  0.8507, Adjusted R-squared:  0.8419
## F-statistic: 96.49 on 61 and 1033 DF,  p-value: < 2.2e-16
```

###A12 Suppression des outliers et nouveaux modèles

```
outlierTest(model_rfe)
```

```
##      rstudent unadjusted p-value Bonferroni p
## 720 -8.156373         9.9568e-16   1.0873e-12
## 819 -5.923211         4.2964e-09   4.6917e-06
## 743 -5.195120         2.4645e-07   2.6912e-04
## 814 -4.342329         1.5487e-05   1.6912e-02
```

```
train3f = train3e[-c(720,819,743,814),]

model_rfe_sansoutliers= lm(log(SalePrice)~., data=train3f)

outlierTest(model_rfe_sansoutliers)
```

```
## No Studentized residuals with Bonferroni p < 0.05
## Largest |rstudent|:
##       rstudent unadjusted p-value Bonferroni p
## 1079 -3.986554         7.1779e-05     0.078096

# outliers issue de bonferonni
train3g = train3f[-c(1079),]

#Modèle final
model_rfe_sansoutliersb= lm(log(SalePrice)~., data=train3g)

#Analyse des AIC
AIC=c(extractAIC(model_rf)[2],extractAIC(model_rfe_sansoutliers)[2],extractAIC
(model_rfe_sansoutliersb)[2])
names(AIC)=c('modRFbase','modFR_outliers1','modFR_outliers2')
AIC

##       modRFbase modFR_outliers1 modFR_outliers2
##       -4329.517       -4150.410       -4146.443

#modèle de base sans la 1iere série outliers
y_Tp_MRFd2 = (predict(model_rf, newdata=train3d))
y_tp_MRFd2 = (predict(model_rf, newdata=test3d))
YTMFd2=exp(y_Tp_MRFd2)
YtMFd2=exp(y_tp_MRFd2)
RMSE_T_MRFd2 = c(sqrt(mean((train3d$SalePrice-YTMFd2)^2)))
RMSE_t_MRFd2 = c(sqrt(mean((test3d$SalePrice-YtMFd2)^2)))
```

Comparaison des erreurs par rapport à la moyenne: le meilleur des trois modèles est celui limité à la 1ier serie de retrait d'outliers

```
print("RMSE RF sans outliers 1 train:"); print(RMSE_T_MRFd2, digits=5)

## [1] "RMSE RF sans outliers 1 train:"

## [1] 26127

print("RMSE RF sans outliers 1 train:"); print(RMSE_t_MRFd2, digits=5)

## [1] "RMSE RF sans outliers 1 train:"

## [1] 25514

summary(train$SalePrice)

##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   52500  129250  164000  180284  214000  611657

22946/180934

## [1] 0.1268197
```

###Annexe 13 description des différentes variables prédictives

- SalePrice - the property's sale price in dollars. This is the target variable that you're trying to predict.

- MSSubClass: The building class
- MSZoning: The general zoning classification
- LotFrontage: Linear feet of street connected to property
- LotArea: Lot size in square feet
- Street: Type of road access
- Alley: Type of alley access
- LotShape: General shape of property
- LandContour: Flatness of the property
- Utilities: Type of utilities available
- LotConfig: Lot configuration
- LandSlope: Slope of property
- Neighborhood: Physical locations within Ames city limits
- Condition1: Proximity to main road or railroad
- Condition2: Proximity to main road or railroad (if a second is present)
- BldgType: Type of dwelling
- HouseStyle: Style of dwelling
- OverallQual: Overall material and finish quality
- OverallCond: Overall condition rating
- YearBuilt: Original construction date
- YearRemodAdd: Remodel date
- RoofStyle: Type of roof
- RoofMatl: Roof material
- Exterior1st: Exterior covering on house
- Exterior2nd: Exterior covering on house (if more than one material)
- MasVnrType: Masonry veneer type
- MasVnrArea: Masonry veneer area in square feet
- ExterQual: Exterior material quality
- ExterCond: Present condition of the material on the exterior
- Foundation: Type of foundation
- BsmtQual: Height of the basement
- BsmtCond: General condition of the basement
- BsmtExposure: Walkout or garden level basement walls
- BsmtFinType1: Quality of basement finished area
- BsmtFinSF1: Type 1 finished square feet
- BsmtFinType2: Quality of second finished area (if present)
- BsmtFinSF2: Type 2 finished square feet
- BsmtUnfSF: Unfinished square feet of basement area
- TotalBsmtSF: Total square feet of basement area
- Heating: Type of heating
- HeatingQC: Heating quality and condition
- CentralAir: Central air conditioning
- Electrical: Electrical system
- 1stFlrSF: First Floor square feet

- 2ndFlrSF: Second floor square feet
- LowQualFinSF: Low quality finished square feet (all floors)
- GrLivArea: Above grade (ground) living area square feet
- BsmtFullBath: Basement full bathrooms
- BsmtHalfBath: Basement half bathrooms
- FullBath: Full bathrooms above grade
- HalfBath: Half baths above grade
- Bedroom: Number of bedrooms above basement level
- Kitchen: Number of kitchens
- KitchenQual: Kitchen quality
- TotRmsAbvGrd: Total rooms above grade (does not include bathrooms)
- Functional: Home functionality rating
- Fireplaces: Number of fireplaces
- FireplaceQu: Fireplace quality
- GarageType: Garage location
- GarageYrBlt: Year garage was built
- GarageFinish: Interior finish of the garage
- GarageCars: Size of garage in car capacity
- GarageArea: Size of garage in square feet
- GarageQual: Garage quality
- GarageCond: Garage condition
- PavedDrive: Paved driveway
- WoodDeckSF: Wood deck area in square feet
- OpenPorchSF: Open porch area in square feet
- EnclosedPorch: Enclosed porch area in square feet
- 3SsnPorch: Three season porch area in square feet
- ScreenPorch: Screen porch area in square feet
- PoolArea: Pool area in square feet
- PoolQC: Pool quality
- Fence: Fence quality
- MiscFeature: Miscellaneous feature not covered in other categories
- MiscVal: $Value of miscellaneous feature
- MoSold: Month Sold
- YrSold: Year Sold
- SaleType: Type of sale
- SaleCondition: Condition of sale